

# Namespace AllenNeuralDynamics.AindBehavior Services.DataTypes

## Classes

[DataTypes](#)

[DeserializeFromJson](#)

Deserializes a sequence of JSON strings into data model objects.

[RenderSynchState](#)

[SerializeToJson](#)

Serializes a sequence of data model objects into JSON strings.

[SoftwareEvent](#)

A software event is a generic event that can be used to track any event that occurs in the software.

## Enums

[DataType](#)

[TimestampSource](#)

# Enum DataType

Namespace: [AllenNeuralDynamics.AindBehaviorServices.DataTypes](#)

Assembly: AllenNeuralDynamics.AindBehaviorServices.dll

```
[JsonConverter(typeof(StringEnumConverter))]  
public enum DataType
```

## Fields

[EnumMember(Value = "array")] Array = 4

[EnumMember(Value = "boolean")] Boolean = 2

[EnumMember(Value = "null")] Null = 5

[EnumMember(Value = "number")] Number = 1

[EnumMember(Value = "object")] Object = 3

[EnumMember(Value = "string")] String = 0

# Class DataTypes

Namespace: [AllenNeuralDynamics.AindBehaviorServices.DataTypes](#)

Assembly: AllenNeuralDynamics.AindBehaviorServices.dll

```
public abstract class DataTypes
```

## Inheritance

[object](#) ← DataTypes

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

# Constructors

## DataTypes()

```
protected DataTypes()
```

## DataTypes(DataTypes)

```
protected DataTypes(DataTypes other)
```

## Parameters

other [DataTypes](#)

# Properties

## AindBehaviorServicesPkgVersion

```
[JsonProperty("aind_behavior_services_pkg_version")]
```

```
public string AindBehaviorServicesPkgVersion { get; set; }
```

Property Value

[string](#)

## RenderSynchState

```
[JsonProperty("render_synch_state", Required = Required.Always)]  
public RenderSynchState RenderSynchState { get; set; }
```

Property Value

[RenderSynchState](#)

## SoftwareEvent

```
[JsonProperty("software_event", Required = Required.Always)]  
public SoftwareEvent SoftwareEvent { get; set; }
```

Property Value

[SoftwareEvent](#)

## Version

```
[JsonProperty("version")]  
public string Version { get; set; }
```

Property Value

[string](#)

## Methods

## PrintMembers(StringBuilder)

```
protected virtual bool PrintMembers(StringBuilder stringBuilder)
```

Parameters

`stringBuilder` [StringBuilder](#)

Returns

`bool`

## ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

`string`

A string that represents the current object.

# Class DeserializeFromJson

Namespace: [AllenNeuralDynamics.AindBehaviorServices.DataTypes](#)

Assembly: AllenNeuralDynamics.AindBehaviorServices.dll

Deserializes a sequence of JSON strings into data model objects.

```
[WorkflowElementCategory(ElementCategory.Transform)]
public class DeserializeFromJson : SingleArgumentExpressionBuilder, IExpressionBuilder
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [SingleArgumentExpressionBuilder](#) ← DeserializeFromJson

## Implements

[IExpressionBuilder](#)

## Inherited Members

[SingleArgumentExpressionBuilder.ArgumentRange](#) , [ExpressionBuilder.ToString\(\)](#) ,  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#) ,  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#) ,  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#) ,  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#) ,  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#) ,  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#) ,  
[ExpressionBuilder.GetElementDisplayName\(object\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### DeserializeFromJson()

```
public DeserializeFromJson()
```

## Properties

## Type

```
public TypeMapping Type { get; set; }
```

### Property Value

[TypeMapping](#)

## Methods

### Build(IEnumerable<Expression>)

Constructs an [Expression](#) node from a collection of input arguments. The result can be chained with other builders in a workflow.

```
public override Expression Build(IEnumerable<Expression> arguments)
```

#### Parameters

**arguments** [IEnumerable](#)<[Expression](#)>

A collection of [Expression](#) nodes representing the input arguments.

#### Returns

[Expression](#)

The constructed [Expression](#) node.

# Class RenderSynchState

Namespace: [AllenNeuralDynamics.AindBehaviorServices.DataTypes](#)

Assembly: AllenNeuralDynamics.AindBehaviorServices.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Source)]
public class RenderSynchState
```

## Inheritance

[object](#) ← RenderSynchState

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### RenderSynchState()

```
public RenderSynchState()
```

### RenderSynchState(RenderSynchState)

```
protected RenderSynchState(RenderSynchState other)
```

## Parameters

other [RenderSynchState](#)

## Properties

### FrameIndex

The frame index of the event

```
[JsonProperty("frame_index")]
public int? FrameIndex { get; set; }
```

Property Value

[int](#)?

## FrameTimestamp

The timestamp of the frame

```
[JsonProperty("frame_timestamp")]
public double? FrameTimestamp { get; set; }
```

Property Value

[double](#)?

## SyncQuadValue

The synchronization quad value

```
[JsonProperty("sync_quad_value")]
public double? SyncQuadValue { get; set; }
```

Property Value

[double](#)?

## Methods

### PrintMembers(StringBuilder)

```
protected virtual bool PrintMembers(StringBuilder stringBuilder)
```

Parameters

**stringBuilder** [StringBuilder](#)

Returns

[bool](#)

## Process()

```
public IObservable<RenderSynchState> Process()
```

Returns

[IObservable](#)<[RenderSynchState](#)>

## Process<TSource>(IObservable<TSource>)

```
public IObservable<RenderSynchState> Process<TSource>(IObservable<TSource> source)
```

Parameters

**source** [IObservable](#)<TSource>

Returns

[IObservable](#)<[RenderSynchState](#)>

## Type Parameters

**TSource**

## ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

string ↗

A string that represents the current object.

# Class SerializeToJson

Namespace: [AllenNeuralDynamics.AindBehaviorServices.DataTypes](#)

Assembly: AllenNeuralDynamics.AindBehaviorServices.dll

Serializes a sequence of data model objects into JSON strings.

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Transform)]
public class SerializeToJson
```

## Inheritance

[object](#) ← SerializeToJson

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Methods

### Process(IObservable<DataTypes>)

```
public IObservable<string> Process(IObservable<DataTypes> source)
```

#### Parameters

source [IObservable](#)<[DataTypes](#)>

#### Returns

[IObservable](#)<[string](#)>

### Process(IObservable<RenderSynchState>)

```
public IObservable<string> Process(IObservable<RenderSynchState> source)
```

Parameters

source [IObservable<RenderSyncState>](#)

Returns

[IObservable<string>](#)

Process([IObservable<SoftwareEvent>](#))

`public IObservable<string> Process(IObservable<SoftwareEvent> source)`

Parameters

source [IObservable<SoftwareEvent>](#)

Returns

[IObservable<string>](#)

# Class SoftwareEvent

Namespace: [AllenNeuralDynamics.AindBehaviorServices.DataTypes](#)

Assembly: AllenNeuralDynamics.AindBehaviorServices.dll

A software event is a generic event that can be used to track any event that occurs in the software.

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Source)]
public class SoftwareEvent
```

## Inheritance

[object](#) ← SoftwareEvent

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### SoftwareEvent()

```
public SoftwareEvent()
```

### SoftwareEvent(SoftwareEvent)

```
protected SoftwareEvent(SoftwareEvent other)
```

## Parameters

other [SoftwareEvent](#)

## Properties

## Data

The data of the event

```
[JsonProperty("data")]
public object Data { get; set; }
```

Property Value

[object](#)

## DataType

The data type of the event

```
[JsonProperty("dataType")]
public DataType DataType { get; set; }
```

Property Value

[DataType](#)

## DataTypeHint

The data type hint of the event

```
[JsonProperty("data_type_hint")]
public string DataTypeHint { get; set; }
```

Property Value

[string](#)

## FrameIndex

The frame index of the event

```
[JsonProperty("frame_index")]
public int? FrameIndex { get; set; }
```

Property Value

[int](#)?

## FrameTimestamp

The timestamp of the frame

```
[JsonProperty("frame_timestamp")]
public double? FrameTimestamp { get; set; }
```

Property Value

[double](#)?

## Name

The name of the event

```
[JsonProperty("name", Required = Required.Always)]
public string Name { get; set; }
```

Property Value

[string](#)

## Timestamp

The timestamp of the event

```
[JsonProperty("timestamp")]
public double? Timestamp { get; set; }
```

Property Value

[double](#)?

## TimestampSource

The source of the timestamp

```
[JsonProperty("timestamp_source")]
public TimestampSource TimestampSource { get; set; }
```

Property Value

[TimestampSource](#)

## Methods

### PrintMembers(StringBuilder)

```
protected virtual bool PrintMembers(StringBuilder stringBuilder)
```

Parameters

[stringBuilder](#) [StringBuilder](#)

Returns

[bool](#)

### Process()

```
public IObservable<SoftwareEvent> Process()
```

Returns

[IObservable](#)<[SoftwareEvent](#)>

## Process<TSource>(IObservable<TSource>)

```
public IObservable<SoftwareEvent> Process<TSource>(IObservable<TSource> source)
```

### Parameters

source [IObservable](#)<TSource>

### Returns

[IObservable](#)<SoftwareEvent>

### Type Parameters

TSource

## ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

### Returns

[string](#)

A string that represents the current object.

# Enum TimestampSource

Namespace: [AllenNeuralDynamics.AindBehaviorServices.DataTypes](#)

Assembly: AllenNeuralDynamics.AindBehaviorServices.dll

```
[JsonConverter(typeof(StringEnumConverter))]  
public enum TimestampSource
```

## Fields

```
[EnumMember(Value = "harp")] Harp = 1
```

```
[EnumMember(Value = "null")] Null = 0
```

```
[EnumMember(Value = "render")] Render = 2
```

# Namespace AllenNeuralDynamics.AindManipulator

## Classes

[AindManipulatorCalibration](#)

Aind manipulator calibration class

[AindManipulatorCalibrationInput](#)

[AindManipulatorCalibrationOutput](#)

[AindManipulatorDevice](#)

[AindManipulatorPosition](#)

[AxisConfiguration](#)

Axis configuration

[BaseModel](#)

[CalibrationRig](#)

[ConfigureMotor](#)

[ContainsKey](#)

[DefaultManipulatorSettings](#)

[DeserializeFromJson](#)

Deserializes a sequence of JSON strings into data model objects.

[HomeAxis](#)

[ManipulatorPosition](#)

[ManipulatorSiUnitConverter](#)

[ModifyMaskMotorRegister](#)

[MoveAbsoluteSingleAxis](#)

[OffsetMotorRegister](#)

[SerializeToJson](#)

Serializes a sequence of data model objects into JSON strings.

## Enums

### [Axis](#)

Motor axis available

### [ConverterMode](#)

### [MicrostepResolution](#)

### [MotorOperationMode](#)

# Class AindManipulatorCalibration

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

Aind manipulator calibration class

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Source)]
public class AindManipulatorCalibration
```

Inheritance

[object](#) ← AindManipulatorCalibration

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

AindManipulatorCalibration()

```
public AindManipulatorCalibration()
```

AindManipulatorCalibration(AindManipulatorCalibration)

```
protected AindManipulatorCalibration(AindManipulatorCalibration other)
```

Parameters

other [AindManipulatorCalibration](#)

## Properties

## Date

```
[JsonProperty("date")]
public DateTimeOffset? Date { get; set; }
```

### Property Value

[DateTimeOffset](#)?

## Description

```
[JsonProperty("description")]
public string Description { get; set; }
```

### Property Value

[string](#)

## DeviceName

Must match a device name in rig/instrument

```
[JsonProperty("device_name")]
public string DeviceName { get; set; }
```

### Property Value

[string](#)

## Input

```
[JsonProperty("input", Required = Required.Always)]
public AindManipulatorCalibrationInput Input { get; set; }
```

### Property Value

## [AindManipulatorCalibrationInput](#)

### Notes

```
[JsonProperty("notes")]
public string Notes { get; set; }
```

### Property Value

[string](#)

### Output

```
[JsonProperty("output", Required = Required.Always)]
public AindManipulatorCalibrationOutput Output { get; set; }
```

### Property Value

[AindManipulatorCalibrationOutput](#)

## Methods

### PrintMembers(StringBuilder)

```
protected virtual bool PrintMembers(StringBuilder stringBuilder)
```

### Parameters

[stringBuilder](#) [StringBuilder](#)

### Returns

[bool](#)

## Process()

```
public IObservable<AindManipulatorCalibration> Process()
```

Returns

[IObservable](#)<[AindManipulatorCalibration](#)>

## Process<TSource>(IObservable<TSource>)

```
public IObservable<AindManipulatorCalibration> Process<TSource>(IObservable<TSource> source)
```

Parameters

source [IObservable](#)<TSource>

Returns

[IObservable](#)<[AindManipulatorCalibration](#)>

Type Parameters

TSource

## ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

[string](#)

A string that represents the current object.

# Class AindManipulatorCalibrationInput

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Source)]
public class AindManipulatorCalibrationInput
```

## Inheritance

[object](#) ← AindManipulatorCalibrationInput

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### AindManipulatorCalibrationInput()

```
public AindManipulatorCalibrationInput()
```

### AindManipulatorCalibrationInput(AindManipulatorCalibrationInput)

```
protected AindManipulatorCalibrationInput(AindManipulatorCalibrationInput other)
```

## Parameters

other [AindManipulatorCalibrationInput](#)

## Properties

### AxisConfiguration

```
[JsonProperty("axis_configuration")]
public List<AxisConfiguration> AxisConfiguration { get; set; }
```

Property Value

[List ↗ <AxisConfiguration>](#)

## FullStepToMm

```
[JsonProperty("full_step_to_mm")]
public ManipulatorPosition FullStepToMm { get; set; }
```

Property Value

[ManipulatorPosition](#)

## HomingOrder

```
[JsonProperty("homing_order")]
public List<Axis> HomingOrder { get; set; }
```

Property Value

[List ↗ <Axis>](#)

## InitialPosition

```
[JsonProperty("initial_position")]
public ManipulatorPosition InitialPosition { get; set; }
```

Property Value

[ManipulatorPosition](#)

# Methods

## PrintMembers(StringBuilder)

```
protected virtual bool PrintMembers(StringBuilder stringBuilder)
```

### Parameters

stringBuilder [StringBuilder](#)

### Returns

[bool](#)

## Process()

```
public IObservable<AindManipulatorCalibrationInput> Process()
```

### Returns

[IObservable](#)<[AindManipulatorCalibrationInput](#)>

## Process<TSource>(IObservable<TSource>)

```
public IObservable<AindManipulatorCalibrationInput> Process<TSource>
(IObservable<TSource> source)
```

### Parameters

source [IObservable](#)<TSource>

### Returns

[IObservable](#)<[AindManipulatorCalibrationInput](#)>

### Type Parameters

## ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

[string](#)

A string that represents the current object.

# Class AindManipulatorCalibrationOutput

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Source)]
public class AindManipulatorCalibrationOutput
```

## Inheritance

[object](#) ← AindManipulatorCalibrationOutput

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### AindManipulatorCalibrationOutput()

```
public AindManipulatorCalibrationOutput()
```

### AindManipulatorCalibrationOutput(AindManipulatorCalibrationOutput)

```
protected AindManipulatorCalibrationOutput(AindManipulatorCalibrationOutput other)
```

## Parameters

other [AindManipulatorCalibrationOutput](#)

## Methods

### PrintMembers(StringBuilder)

```
protected virtual bool PrintMembers(StringBuilder stringBuilder)
```

Parameters

stringBuilder [StringBuilder](#)

Returns

[bool](#)

## Process()

```
public IObservable<AindManipulatorCalibrationOutput> Process()
```

Returns

[IObservable](#) <[AindManipulatorCalibrationOutput](#)>

## Process<TSource>(IObservable<TSource>)

```
public IObservable<AindManipulatorCalibrationOutput> Process<TSource>(IObservable<TSource> source)
```

Parameters

source [IObservable](#) <TSource>

Returns

[IObservable](#) <[AindManipulatorCalibrationOutput](#)>

## Type Parameters

TSource

## ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

string ↗

A string that represents the current object.

# Class AindManipulatorDevice

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Source)]
public class AindManipulatorDevice
```

## Inheritance

[object](#) ← AindManipulatorDevice

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### AindManipulatorDevice()

```
public AindManipulatorDevice()
```

### AindManipulatorDevice(AindManipulatorDevice)

```
protected AindManipulatorDevice(AindManipulatorDevice other)
```

## Parameters

other [AindManipulatorDevice](#)

## Properties

### AdditionalSettings

Additional settings

```
[JsonProperty("additional_settings")]
public BaseModel AdditionalSettings { get; set; }
```

Property Value

[BaseModel](#)

## Calibration

```
[JsonProperty("calibration")]
public AindManipulatorCalibration Calibration { get; set; }
```

Property Value

[AindManipulatorCalibration](#)

## DeviceType

```
[JsonProperty("device_type")]
public string DeviceType { get; set; }
```

Property Value

[string](#) ↗

## PortName

Device port name

```
[JsonProperty("port_name", Required = Required.Always)]
public string PortName { get; set; }
```

Property Value

[string](#) ↗

## SerialNumber

Device serial number

```
[JsonProperty("serial_number")]
public string SerialNumber { get; set; }
```

Property Value

[string](#)

## WhoAmI

```
[JsonProperty("who_am_i")]
public int WhoAmI { get; set; }
```

Property Value

[int](#)

## Methods

### PrintMembers(StringBuilder)

```
protected virtual bool PrintMembers(StringBuilder stringBuilder)
```

Parameters

[stringBuilder](#) [StringBuilder](#)

Returns

[bool](#)

## Process()

```
public IObservable<AindManipulatorDevice> Process()
```

Returns

[IObservable](#) <[AindManipulatorDevice](#)>

## Process<TSource>(IObservable<TSource>)

```
public IObservable<AindManipulatorDevice> Process<TSource>(IObservable<TSource> source)
```

Parameters

source [IObservable](#) <TSource>

Returns

[IObservable](#) <[AindManipulatorDevice](#)>

Type Parameters

TSource

## ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

[string](#)

A string that represents the current object.

# Class AindManipulatorPosition

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

```
public class AindManipulatorPosition
```

## Inheritance

[object](#) ← AindManipulatorPosition

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### AindManipulatorPosition()

```
public AindManipulatorPosition()
```

### AindManipulatorPosition(ManipulatorPosition)

```
public AindManipulatorPosition(ManipulatorPosition pos)
```

#### Parameters

pos [ManipulatorPosition](#)

### AindManipulatorPosition(double, double, double, double)

```
public AindManipulatorPosition(double y1, double y2, double x, double z)
```

#### Parameters

y1 [double](#)

y2 [double](#)

x [double](#)

z [double](#)

## Fields

X

`public double X`

Field Value

[double](#)

Y1

`public double Y1`

Field Value

[double](#)

Y2

`public double Y2`

Field Value

[double](#)

Z

```
public double Z
```

Field Value

[double](#)

## Methods

### Equals(AindManipulatorPosition, AindManipulatorPosition)

```
public static bool Equals(AindManipulatorPosition e11, AindManipulatorPosition e12)
```

Parameters

e11 [AindManipulatorPosition](#)

e12 [AindManipulatorPosition](#)

Returns

[bool](#)

### Equals(object)

Determines whether the specified object is equal to the current object.

```
public override bool Equals(object obj)
```

Parameters

obj [object](#)

The object to compare with the current object.

Returns

[bool](#)

[true](#) if the specified object is equal to the current object; otherwise, [false](#).

## GetHashCode()

Serves as the default hash function.

```
public override int GetHashCode()
```

Returns

[int](#)

A hash code for the current object.

## ToManipulatorPosition()

```
public ManipulatorPosition ToManipulatorPosition()
```

Returns

[ManipulatorPosition](#)

## Operators

operator ==(AindManipulatorPosition,  
AindManipulatorPosition)

```
public static bool operator ==(AindManipulatorPosition x, AindManipulatorPosition y)
```

Parameters

x [AindManipulatorPosition](#)

y [AindManipulatorPosition](#)

Returns

[bool](#) ↗

## operator !=(AindManipulatorPosition, AindManipulatorPosition)

```
public static bool operator !=(AindManipulatorPosition x, AindManipulatorPosition y)
```

### Parameters

x [AindManipulatorPosition](#)

y [AindManipulatorPosition](#)

### Returns

[bool](#) ↗

# Enum Axis

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

Motor axis available

```
public enum Axis
```

## Fields

```
[EnumMember(Value = "0")] None = 0
```

```
[EnumMember(Value = "1")] X = 1
```

```
[EnumMember(Value = "2")] Y1 = 2
```

```
[EnumMember(Value = "3")] Y2 = 3
```

```
[EnumMember(Value = "4")] Z = 4
```

# Class AxisConfiguration

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

Axis configuration

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Source)]
public class AxisConfiguration
```

Inheritance

[object](#) ← AxisConfiguration

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

AxisConfiguration()

```
public AxisConfiguration()
```

AxisConfiguration(AxisConfiguration)

```
protected AxisConfiguration(AxisConfiguration other)
```

Parameters

other [AxisConfiguration](#)

## Properties

## Axis

```
[JsonProperty("axis", Required = Required.Always)]  
public Axis Axis { get; set; }
```

Property Value

[Axis](#)

## MaxLimit

```
[JsonProperty("max_limit")]  
public double MaxLimit { get; set; }
```

Property Value

[double](#) ↗

## MaximumStepInterval

```
[JsonProperty("maximum_step_interval")]  
public int MaximumStepInterval { get; set; }
```

Property Value

[int](#) ↗

## MicrostepResolution

```
[JsonProperty("microstep_resolution")]  
public MicrostepResolution MicrostepResolution { get; set; }
```

Property Value

[MicrostepResolution](#)

## MinLimit

```
[JsonProperty("min_limit")]
public double MinLimit { get; set; }
```

Property Value

[double](#) ↗

## MotorOperationMode

```
[JsonProperty("motor_operation_mode")]
public MotorOperationMode MotorOperationMode { get; set; }
```

Property Value

[MotorOperationMode](#)

## StepAccelerationInterval

Acceleration of the step interval in microseconds

```
[JsonProperty("step_acceleration_interval")]
public int StepAccelerationInterval { get; set; }
```

Property Value

[int](#) ↗

## StepInterval

Step interval in microseconds.

```
[JsonProperty("step_interval")]
public int StepInterval { get; set; }
```

Property Value

[int](#)

## Methods

PrintMembers(StringBuilder)

```
protected virtual bool PrintMembers(StringBuilder stringBuilder)
```

Parameters

stringBuilder [StringBuilder](#)

Returns

[bool](#)

Process()

```
public IObservable<AxisConfiguration> Process()
```

Returns

[IObservable](#)<[AxisConfiguration](#)>

Process<TSource>(IObservable<TSource>)

```
public IObservable<AxisConfiguration> Process<TSource>(IObservable<TSource> source)
```

Parameters

source [IObservable](#)<TSource>

Returns

[IObservable](#)<AxisConfiguration>

Type Parameters

TSource

## ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

[string](#)

A string that represents the current object.

# Class BaseModel

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Source)]
public class BaseModel
```

## Inheritance

[object](#) ← BaseModel

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### BaseModel()

```
public BaseModel()
```

### BaseModel(BaseModel)

```
protected BaseModel(BaseModel other)
```

## Parameters

other [BaseModel](#)

## Methods

### PrintMembers(StringBuilder)

```
protected virtual bool PrintMembers(StringBuilder stringBuilder)
```

Parameters

**stringBuilder** [StringBuilder](#)

Returns

[bool](#)

## Process()

```
public IObservable<BaseModel> Process()
```

Returns

[IObservable](#) <[BaseModel](#)>

## Process<TSource>(IObservable<TSource>)

```
public IObservable<BaseModel> Process<TSource>(IObservable<TSource> source)
```

Parameters

**source** [IObservable](#) <TSource>

Returns

[IObservable](#) <[BaseModel](#)>

## Type Parameters

**TSource**

## ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

[string](#)

A string that represents the current object.

# Class CalibrationRig

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Source)]
public class CalibrationRig
```

## Inheritance

[object](#) ← CalibrationRig

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### CalibrationRig()

```
public CalibrationRig()
```

### CalibrationRig(CalibrationRig)

```
protected CalibrationRig(CalibrationRig other)
```

## Parameters

other [CalibrationRig](#)

## Properties

### AindBehaviorServicesPkgVersion

```
[JsonProperty("aind_behavior_services_pkg_version")]
public string AindBehaviorServicesPkgVersion { get; set; }
```

Property Value

[string](#)

## ComputerName

Computer name

```
[JsonProperty("computer_name")]
public string ComputerName { get; set; }
```

Property Value

[string](#)

## Manipulator

```
[JsonProperty("manipulator")]
public AindManipulatorDevice Manipulator { get; set; }
```

Property Value

[AindManipulatorDevice](#)

## RigName

Rig name

```
[JsonProperty("rig_name", Required = Required.Always)]
public string RigName { get; set; }
```

Property Value

[string](#)

## Version

```
[JsonProperty("version")]
public string Version { get; set; }
```

Property Value

[string](#)

## Methods

### PrintMembers(StringBuilder)

```
protected virtual bool PrintMembers(StringBuilder stringBuilder)
```

Parameters

[stringBuilder](#) [StringBuilder](#)

Returns

[bool](#)

### Process()

```
public IObservable<CalibrationRig> Process()
```

Returns

[IObservable](#) <[CalibrationRig](#)>

### Process<TSource>(IObservable<TSource>)

```
public IObservable<CalibrationRig> Process<TSource>(IObservable<TSource> source)
```

Parameters

source [IObservable](#)<TSource>

Returns

[IObservable](#)<CalibrationRig>

Type Parameters

TSource

## ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

[string](#)

A string that represents the current object.

# Class ConfigureMotor

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

```
public static class ConfigureMotor
```

## Inheritance

[object](#) ← ConfigureMotor

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Class ContainsKey

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Transform)]
public class ContainsKey
```

## Inheritance

[object](#) ← ContainsKey

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Methods

Process<TKey, TValue>(IObservable<Tuple<IDictionary<TKey, TValue>, TKey>>)

```
public IObservable<bool> Process<TKey, TValue>(IObservable<Tuple<IDictionary<TKey, TValue>, TKey>> source)
```

## Parameters

source [IObservable](#)<[Tuple](#)<[IDictionary](#)<TKey, TValue>, TKey>>

## Returns

[IObservable](#)<[bool](#)>

## Type Parameters

TKey

TValue

`Process<TKey, TValue>(IObservable<Tuple<TKey, IDictionary<TKey, TValue>>>)`

```
public IObservable<bool> Process<TKey, TValue>(IObservable<Tuple<TKey, IDictionary<TKey, TValue>>> source)
```

Parameters

`source` [IObservable<Tuple<TKey, IDictionary<TKey, TValue>>>](#)

Returns

[IObservable<bool>](#)

Type Parameters

`TKey`

`TValue`

# Enum ConverterMode

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

```
public enum ConverterMode
```

## Fields

MmToStep = 1

StepToMm = 0

# Class DefaultManipulatorSettings

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

```
public class DefaultManipulatorSettings : Source<AindManipulatorCalibrationInput>
```

## Inheritance

[object](#) ← [Source](#) <[AindManipulatorCalibrationInput](#)> ← DefaultManipulatorSettings

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Properties

### EnabledAxis

```
[TypeConverter(typeof(UnidimensionalArrayConverter))]  
public Axis[] EnabledAxis { get; set; }
```

### Property Value

[Axis\[\]](#)

### FullStepToMm

```
[TypeConverter(typeof(NumericRecordConverter))]  
public AindManipulatorPosition FullStepToMm { get; set; }
```

### Property Value

[AindManipulatorPosition](#)

## HomingOrder

```
[TypeConverter(typeof(UnidimensionalArrayConverter))]  
public Axis[] HomingOrder { get; set; }
```

Property Value

[Axis\[\]](#)

## InitialPosition

```
[TypeConverter(typeof(NumericRecordConverter))]  
public AindManipulatorPosition InitialPosition { get; set; }
```

Property Value

[AindManipulatorPosition](#)

## MaxLimit

```
public double MaxLimit { get; set; }
```

Property Value

[double](#)

## MaximumStepInterval

```
public int MaximumStepInterval { get; set; }
```

Property Value

[int](#)

## MicrostepResolution

```
public MicrostepResolution MicrostepResolution { get; set; }
```

Property Value

[MicrostepResolution](#) ↗

## MinLimit

```
public double MinLimit { get; set; }
```

Property Value

[double](#) ↗

## MotorOperationMode

```
public MotorOperationMode MotorOperationMode { get; set; }
```

Property Value

[MotorOperationMode](#) ↗

## StepAccelerationInterval

```
public int StepAccelerationInterval { get; set; }
```

Property Value

[int](#) ↗

## StepInterval

```
public int StepInterval { get; set; }
```

Property Value

[int](#)

## Methods

### Generate()

Generates an observable sequence of data elements.

```
public override IObservable<AindManipulatorCalibrationInput> Generate()
```

Returns

[IObservable](#) <[AindManipulatorCalibrationInput](#)>

An observable sequence of data elements of type [AindManipulatorCalibrationInput](#).

# Class DeserializeFromJson

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

Deserializes a sequence of JSON strings into data model objects.

```
[WorkflowElementCategory(ElementCategory.Transform)]
public class DeserializeFromJson : SingleArgumentExpressionBuilder, IExpressionBuilder
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [SingleArgumentExpressionBuilder](#) ← DeserializeFromJson

## Implements

[IExpressionBuilder](#)

## Inherited Members

[SingleArgumentExpressionBuilder.ArgumentRange](#) , [ExpressionBuilder.ToString\(\)](#) ,  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#) ,  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#) ,  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#) ,  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#) ,  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#) ,  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#) ,  
[ExpressionBuilder.GetElementDisplayName\(object\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### DeserializeFromJson()

```
public DeserializeFromJson()
```

## Properties

## Type

```
public TypeMapping Type { get; set; }
```

### Property Value

[TypeMapping](#)

## Methods

### Build(IEnumerable<Expression>)

Constructs an [Expression](#) node from a collection of input arguments. The result can be chained with other builders in a workflow.

```
public override Expression Build(IEnumerable<Expression> arguments)
```

#### Parameters

**arguments** [IEnumerable](#)<[Expression](#)>

A collection of [Expression](#) nodes representing the input arguments.

#### Returns

[Expression](#)

The constructed [Expression](#) node.

# Class HomeAxis

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

```
public static class HomeAxis
```

## Inheritance

[object](#) ← HomeAxis

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Class ManipulatorPosition

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Source)]
public class ManipulatorPosition
```

## Inheritance

[object](#) ← ManipulatorPosition

## Inherited Members

[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Constructors

### ManipulatorPosition()

```
public ManipulatorPosition()
```

### ManipulatorPosition(ManipulatorPosition)

```
protected ManipulatorPosition(ManipulatorPosition other)
```

## Parameters

other [ManipulatorPosition](#)

## Properties

### this[Axis]

```
public double this[Axis axis] { get; set; }
```

Parameters

axis [Axis](#)

Property Value

[double](#) ↗

this[int]

```
public double this[int axisIndex] { get; set; }
```

Parameters

axisIndex [int](#) ↗

Property Value

[double](#) ↗

X

```
[JsonProperty("x", Required = Required.Always)]  
public double X { get; set; }
```

Property Value

[double](#) ↗

Y1

```
[JsonProperty("y1", Required = Required.Always)]  
public double Y1 { get; set; }
```

Property Value

[double ↗](#)

Y2

```
[JsonProperty("y2", Required = Required.Always)]  
public double Y2 { get; set; }
```

Property Value

[double ↗](#)

Z

```
[JsonProperty("z", Required = Required.Always)]  
public double Z { get; set; }
```

Property Value

[double ↗](#)

## Methods

Equals(ManipulatorPosition, ManipulatorPosition)

```
public static bool Equals(ManipulatorPosition e1, ManipulatorPosition e2)
```

Parameters

e11 [ManipulatorPosition](#)

e12 [ManipulatorPosition](#)

Returns

[bool](#)

## Equals(object)

Determines whether the specified object is equal to the current object.

```
public override bool Equals(object obj)
```

Parameters

[obj](#) [object](#)

The object to compare with the current object.

Returns

[bool](#)

[true](#) if the specified object is equal to the current object; otherwise, [false](#).

## GetHashCode()

Serves as the default hash function.

```
public override int GetHashCode()
```

Returns

[int](#)

A hash code for the current object.

## PrintMembers(StringBuilder)

```
protected virtual bool PrintMembers(StringBuilder stringBuilder)
```

Parameters

`stringBuilder` [StringBuilder](#)

Returns

[bool](#)

## Process()

`public Iobservable<ManipulatorPosition> Process()`

Returns

[Iobservable](#)<[ManipulatorPosition](#)>

## Process<TSource>(Iobservable<TSource>)

`public Iobservable<ManipulatorPosition> Process<TSource>(Iobservable<TSource> source)`

Parameters

`source` [Iobservable](#)<[TSource](#)>

Returns

[Iobservable](#)<[ManipulatorPosition](#)>

Type Parameters

`TSource`

## ToString()

Returns a string that represents the current object.

`public override string ToString()`

Returns

[string](#)

A string that represents the current object.

## Operators

### operator +(ManipulatorPosition, ManipulatorPosition)

```
public static ManipulatorPosition operator +(ManipulatorPosition e11,  
ManipulatorPosition e12)
```

Parameters

e11 [ManipulatorPosition](#)

e12 [ManipulatorPosition](#)

Returns

[ManipulatorPosition](#)

### operator /(ManipulatorPosition, ManipulatorPosition)

```
public static ManipulatorPosition operator /(ManipulatorPosition e11,  
ManipulatorPosition e12)
```

Parameters

e11 [ManipulatorPosition](#)

e12 [ManipulatorPosition](#)

Returns

[ManipulatorPosition](#)

## operator ==(ManipulatorPosition, ManipulatorPosition)

```
public static bool operator ==(ManipulatorPosition x, ManipulatorPosition y)
```

Parameters

x [ManipulatorPosition](#)

y [ManipulatorPosition](#)

Returns

[bool](#) ↗

## operator !=(ManipulatorPosition, ManipulatorPosition)

```
public static bool operator !=(ManipulatorPosition x, ManipulatorPosition y)
```

Parameters

x [ManipulatorPosition](#)

y [ManipulatorPosition](#)

Returns

[bool](#) ↗

## operator \*(ManipulatorPosition, ManipulatorPosition)

```
public static ManipulatorPosition operator *(ManipulatorPosition el1,  
ManipulatorPosition el2)
```

Parameters

el1 [ManipulatorPosition](#)

e12 [ManipulatorPosition](#)

Returns

[ManipulatorPosition](#)

operator \*(ManipulatorPosition, float)

```
public static ManipulatorPosition operator *(ManipulatorPosition e11, float gain)
```

Parameters

e11 [ManipulatorPosition](#)

gain [float](#) ↗

Returns

[ManipulatorPosition](#)

operator -(ManipulatorPosition, ManipulatorPosition)

```
public static ManipulatorPosition operator -(ManipulatorPosition e11,
ManipulatorPosition e12)
```

Parameters

e11 [ManipulatorPosition](#)

e12 [ManipulatorPosition](#)

Returns

[ManipulatorPosition](#)

# Class ManipulatorSiUnitConverter

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Transform)]
public class ManipulatorSiUnitConverter
```

## Inheritance

[object](#) ← ManipulatorSiUnitConverter

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Properties

### Calibration

```
public AindManipulatorCalibrationInput Calibration { get; set; }
```

### Property Value

[AindManipulatorCalibrationInput](#)

### ConverterMode

```
public ConverterMode ConverterMode { get; set; }
```

### Property Value

[ConverterMode](#)

# Methods

## Process(I<sup>0</sup>bservable<ManipulatorPosition>)

```
public I0bservable<ManipulatorPosition> Process(I0bservable<ManipulatorPosition> source)
```

### Parameters

source [I<sup>0</sup>bservable<ManipulatorPosition>](#)

### Returns

[I<sup>0</sup>bservable<ManipulatorPosition>](#)

## Process(I<sup>0</sup>bservable<Tuple<Axis, double>>)

```
public I0bservable<double> Process(I0bservable<Tuple<Axis, double>> source)
```

### Parameters

source [I<sup>0</sup>bservable<Tuple<Axis, double>>](#)

### Returns

[I<sup>0</sup>bservable<double>](#)

## Process(I<sup>0</sup>bservable<Tuple<Axis, int>>)

```
public I0bservable<double> Process(I0bservable<Tuple<Axis, int>> source)
```

### Parameters

source [I<sup>0</sup>bservable<Tuple<Axis, int>>](#)

### Returns

[IObservable<double>](#)

# Enum MicrostepResolution

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

```
public enum MicrostepResolution
```

## Fields

```
[EnumMember(Value = "1")] Microstep16 = 1
```

```
[EnumMember(Value = "2")] Microstep32 = 2
```

```
[EnumMember(Value = "3")] Microstep64 = 3
```

```
[EnumMember(Value = "0")] Microstep8 = 0
```

# Class ModifyMaskMotorRegister

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

```
public static class ModifyMaskMotorRegister
```

## Inheritance

[object](#) ← ModifyMaskMotorRegister

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Enum MotorOperationMode

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

```
public enum MotorOperationMode
```

## Fields

```
[EnumMember(Value = "1")] Dynamic = 1
```

```
[EnumMember(Value = "0")] Quiet = 0
```

# Class MoveAbsoluteSingleAxis

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Combinator)]
public class MoveAbsoluteSingleAxis
```

## Inheritance

[object](#) ← MoveAbsoluteSingleAxis

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Axis

```
public Axis? Axis { get; set; }
```

## Property Value

[Axis?](#)

# Methods

## Process(IObservable<AindManipulatorPosition>)

```
public IObservable<HarpMessage> Process(IObservable<AindManipulatorPosition> source)
```

## Parameters

source [IObservable<AindManipulatorPosition>](#)

Returns

[IObservable<HarpMessage>](#)

## Process(IObservable<ManipulatorPosition>)

public IObservable<HarpMessage> **Process**(IObservable<ManipulatorPosition> source)

Parameters

source [IObservable<ManipulatorPosition>](#)

Returns

[IObservable<HarpMessage>](#)

## Process(IObservable<int>)

public IObservable<HarpMessage> **Process**(IObservable<int> source)

Parameters

source [IObservable<int>](#)

Returns

[IObservable<HarpMessage>](#)

# Class OffsetMotorRegister

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

```
public static class OffsetMotorRegister
```

## Inheritance

[object](#) ← OffsetMotorRegister

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Class SerializeToJson

Namespace: [AllenNeuralDynamics.AindManipulator](#)

Assembly: AllenNeuralDynamics.AindManipulator.dll

Serializes a sequence of data model objects into JSON strings.

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Transform)]
public class SerializeToJson
```

## Inheritance

[object](#) ← SerializeToJson

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

## Methods

### Process(I<sub>0</sub>bservable<AindManipulatorCalibrationInput>)

```
public I0bservable<string> Process(I0bservable<AindManipulatorCalibrationInput> source)
```

#### Parameters

source [I<sub>0</sub>bservable](#)<[AindManipulatorCalibrationInput](#)>

#### Returns

[I<sub>0</sub>bservable](#)<[string](#)>

### Process(I<sub>0</sub>bservable<AindManipulatorCalibrationOutput>)

```
public I0bservable<string> Process(I0bservable<AindManipulatorCalibrationOutput> source)
```

Parameters

source [IObservable<AindManipulatorCalibrationOutput>](#)

Returns

[IObservable<string>](#)

## Process([IObservable<AindManipulatorCalibration>](#))

```
public IObservable<string> Process(IObservable<AindManipulatorCalibration> source)
```

Parameters

source [IObservable<AindManipulatorCalibration>](#)

Returns

[IObservable<string>](#)

## Process([IObservable<AindManipulatorDevice>](#))

```
public IObservable<string> Process(IObservable<AindManipulatorDevice> source)
```

Parameters

source [IObservable<AindManipulatorDevice>](#)

Returns

[IObservable<string>](#)

## Process([IObservable<AxisConfiguration>](#))

```
public IObservable<string> Process(IObservable<AxisConfiguration> source)
```

Parameters

source [IObservable<AxisConfiguration>](#)

Returns

[IObservable<string>](#)

## Process([IObservable<BaseModel>](#))

```
public IObservable<string> Process(IObservable<BaseModel> source)
```

Parameters

source [IObservable<BaseModel>](#)

Returns

[IObservable<string>](#)

## Process([IObservable<CalibrationRig>](#))

```
public IObservable<string> Process(IObservable<CalibrationRig> source)
```

Parameters

source [IObservable<CalibrationRig>](#)

Returns

[IObservable<string>](#)

## Process([IObservable<ManipulatorPosition>](#))

```
public IObservable<string> Process(IObservable<ManipulatorPosition> source)
```

## Parameters

source [IObservable<ManipulatorPosition>](#)

## Returns

[IObservable<string>](#)

# Namespace AllenNeuralDynamics.Alicat Flowmeter

## Classes

[FlowmeterDataframe](#)

[Parse](#)

# Class FlowmeterDataframe

Namespace: [AllenNeuralDynamics.AlicatFlowmeter](#)

Assembly: AllenNeuralDynamics.AlicatFlowmeter.dll

```
public class FlowmeterDataframe
```

## Inheritance

[object](#) ← FlowmeterDataframe

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### FlowmeterDataframe()

```
public FlowmeterDataframe()
```

### FlowmeterDataframe(string)

```
public FlowmeterDataframe(string serialString)
```

## Parameters

serialString [string](#)

## Properties

### AbsolutePressure

```
public float AbsolutePressure { get; set; }
```

PropertyValue

[float](#)

DeviceId

```
public string DeviceId { get; set; }
```

PropertyValue

[string](#)

Gas

```
public string Gas { get; set; }
```

PropertyValue

[string](#)

MassFlowRate

```
public float MassFlowRate { get; set; }
```

PropertyValue

[float](#)

MassFlowTotal

```
public float MassFlowTotal { get; set; }
```

PropertyValue

[float](#) ↗

## Temperature

```
public float Temperature { get; set; }
```

### Property Value

[float](#) ↗

## VolumetricFlowRate

```
public float VolumetricFlowRate { get; set; }
```

### Property Value

[float](#) ↗

## Methods

### Default()

```
public static FlowmeterDataframe Default()
```

### Returns

[FlowmeterDataframe](#)

### Parse(string)

```
public static FlowmeterDataframe Parse(string value)
```

### Parameters

**value** [string](#)

Returns

[FlowmeterDataframe](#)

## ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

[string](#)

A string that represents the current object.

# Class Parse

Namespace: [AllenNeuralDynamics.AlicatFlowmeter](#)

Assembly: AllenNeuralDynamics.AlicatFlowmeter.dll

```
public class Parse : Transform<string, FlowmeterDataframe>
```

## Inheritance

```
object ↪ Combinator<string ↪ , FlowmeterDataframe> ↪  
Transform<string ↪ , FlowmeterDataframe> ↪ Parse
```

## Inherited Members

```
Combinator<string, FlowmeterDataframe>.Process(IObservable<string>) ↪ , object.ToString() ↪ ,  
object.Equals(object) ↪ , object.Equals(object, object) ↪ , object.ReferenceEquals(object, object) ↪ ,  
object.GetHashCode() ↪ , object.GetType() ↪ , object.MemberwiseClone() ↪
```

# Methods

## Process(IObservable<string>)

Processes an observable sequence into a new sequence of the specified element type.

```
public override IObservable<FlowmeterDataframe> Process(IObservable<string> source)
```

### Parameters

source [IObservable<string>](#)

The source sequence to process.

### Returns

[IObservable<FlowmeterDataframe>](#)

An observable sequence with elements of type [FlowmeterDataframe](#).

# Namespace AllenNeuralDynamics.Core

## Classes

[AindSpinnakerCapture](#)

[AppendModalitySuffix](#)

[CaptureProcess](#)

Represents an operator that starts a new system process with the specified file name and command-line arguments.

[ContainsKey](#)

[CreateSoftwareEvent](#)

[FfmpegVideoWriter](#)

[FileCopy](#)

[FramestampSoftwareEvent](#)

[JsonWriter](#)

[StartProcessOnNewConsole](#)

[StartRobocopy](#)

[StripDirectoryPath](#)

[ThrowException](#)

[TimeSpanFromSeconds](#)

[TimestampSoftwareEvent](#)

[YamlWriter](#)

## Enums

[Modality](#)

[Verbosity](#)

# Class AindSpinnakerCapture

Namespace: [AllenNeuralDynamics.Core](#)

Assembly: AllenNeuralDynamics.Core.dll

```
public class AindSpinnakerCapture : SpinnakerCapture
```

## Inheritance

[object](#) ← [Source](#)<SpinnakerDataFrame> ← SpinnakerCapture ← AindSpinnakerCapture

## Inherited Members

SpinnakerCapture.Generate() , [SpinnakerCapture.Generate<TSource>\(IObservable<TSource>\)](#) ,  
SpinnakerCapture.Index , SpinnakerCapture.SerialNumber , SpinnakerCapture.ColorProcessing ,  
[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Constructors

AindSpinnakerCapture()

```
public AindSpinnakerCapture()
```

## Properties

AdcBitDepth

```
public AdcBitDepthEnums? AdcBitDepth { get; set; }
```

Property Value

AdcBitDepthEnums?

Binning

```
public int Binning { get; set; }
```

Property Value

[int ↗](#)

## ExposureTime

```
public double ExposureTime { get; set; }
```

Property Value

[double ↗](#)

## Gain

```
public double Gain { get; set; }
```

Property Value

[double ↗](#)

## Gamma

```
public double? Gamma { get; set; }
```

Property Value

[double ↗?](#)

## PixelFormat

```
public PixelFormatEnums? PixelFormat { get; set; }
```

Property Value

PixelFormatEnums?

## RegionOfInterest

```
public Rect RegionOfInterest { get; set; }
```

Property Value

[Rect](#)

## Methods

### Configure(IManagedCamera)

```
protected override void Configure(IManagedCamera camera)
```

Parameters

**camera** IManagedCamera

# Class AppendModalitySuffix

Namespace: [AllenNeuralDynamics.Core](#)

Assembly: AllenNeuralDynamics.Core.dll

```
public class AppendModalitySuffix : Transform<string, string>
```

## Inheritance

[object](#) ← [Combinator](#)<[string](#), [string](#)> ← [Transform](#)<[string](#), [string](#)> ← AppendModalitySuffix

## Inherited Members

[Combinator<string, string>.Process\(IObservable<string>\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

# Properties

## Modality

```
public Modality? Modality { get; set; }
```

## Property Value

[Modality?](#)

# Methods

## Process(IObservable<string>)

Processes an observable sequence into a new sequence of the specified element type.

```
public override IObservable<string> Process(IObservable<string> source)
```

## Parameters

source [IObservable](#)<[string](#)>

The source sequence to process.

Returns

[IObservable](#)<[string](#)>

An observable sequence with elements of type [string](#).

# Class CaptureProcess

Namespace: [AllenNeuralDynamics.Core](#)

Assembly: AllenNeuralDynamics.Core.dll

Represents an operator that starts a new system process with the specified file name and command-line arguments.

```
public class CaptureProcess : Source<string>
```

## Inheritance

[object](#) ← [Source](#)<[string](#)> ← CaptureProcess

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Arguments

```
public string Arguments { get; set; }
```

### Property Value

[string](#)

## FileName

```
[FileNameFilter("Executable files|*.exe|All Files|*.*")]
public string FileName { get; set; }
```

### Property Value

[string](#)

# Methods

## Generate()

Generates an observable sequence of data elements.

```
public override IObservable<string> Generate()
```

Returns

[IObservable<string>](#)

An observable sequence of data elements of type [string](#).

## Generate<TSource>(IObservable<TSource>)

```
public IObservable<string> Generate<TSource>(IObservable<TSource> source)
```

Parameters

source [IObservable<TSource>](#)

Returns

[IObservable<string>](#)

Type Parameters

TSource

# Class ContainsKey

Namespace: [AllenNeuralDynamics.Core](#)

Assembly: AllenNeuralDynamics.Core.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Transform)]
public class ContainsKey
```

## Inheritance

[object](#) ← ContainsKey

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Key

```
public string Key { get; set; }
```

## Property Value

[string](#)

# Methods

## Process<TValue>(IObservable<IDictionary<string, TValue>>)

```
public IObservable<bool> Process<TValue>(IObservable<IDictionary<string, TValue>> source)
```

## Parameters

source [IObservable](#)<[IDictionary](#)<[string](#), TValue>>

Returns

[IObservable](#)<bool>

Type Parameters

TValue

# Class CreateSoftwareEvent

Namespace: [AllenNeuralDynamics.Core](#)

Assembly: AllenNeuralDynamics.Core.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Transform)]
public class CreateSoftwareEvent
```

## Inheritance

[object](#) ← CreateSoftwareEvent

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## EventName

```
public string EventName { get; set; }
```

## Property Value

[string](#)

# Methods

## Process<TSource>(IObservable<Timestamped<TSource>>)

```
public IObservable<SoftwareEvent> Process<TSource>(IObservable<Timestamped<TSource>> source)
```

## Parameters

source [IObservable<Timestamped<TSource>>](#)

Returns

[IObservable<SoftwareEvent>](#)

Type Parameters

TSource

Process<TSource>(IObservable<TSource>)

public [IObservable<SoftwareEvent>](#) Process<TSource>(IObservable<TSource> source)

Parameters

source [IObservable<TSource>](#)

Returns

[IObservable<SoftwareEvent>](#)

Type Parameters

TSource

# Class FfmpegVideoWriter

Namespace: [AllenNeuralDynamics.Core](#)

Assembly: AllenNeuralDynamics.Core.dll

```
public class FfmpegVideoWriter : Sink<IplImage>
```

## Inheritance

[object](#) ← [Combinator](#)<[IplImage](#), [IplImage](#)> ← [Sink](#)<[IplImage](#)> ← FfmpegVideoWriter

## Inherited Members

[Combinator<IplImage, IplImage>.Process\(IObservable<IplImage>\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

# Properties

## FileName

```
public string FileName { get; set; }
```

## Property Value

[string](#)

## FrameRate

```
public int FrameRate { get; set; }
```

## Property Value

[int](#)

## InputArguments

```
public string InputArguments { get; set; }
```

Property Value

[string](#)

## OutputArguments

```
public string OutputArguments { get; set; }
```

Property Value

[string](#)

## Overwrite

```
public bool Overwrite { get; set; }
```

Property Value

[bool](#)

## Verbosity

```
public Verbosity Verbosity { get; set; }
```

Property Value

[Verbosity](#)

## Methods

Process(IObservable<IpImage>)

Processes an observable sequence into a new sequence of the specified element type.

```
public override IObservable<IplImage> Process(IObservable<IplImage> source)
```

## Parameters

**source** [IObservable](#)<[IplImage](#)>

The source sequence to process.

## Returns

[IObservable](#)<[IplImage](#)>

An observable sequence with elements of type [IplImage](#).

# Class FileCopy

Namespace: [AllenNeuralDynamics.Core](#)

Assembly: AllenNeuralDynamics.Core.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Sink)]
public class FileCopy
```

## Inheritance

[object](#) ← FileCopy

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## DestinationPath

```
public string DestinationPath { get; set; }
```

### Property Value

[string](#)

## Overwrite

```
public bool Overwrite { get; set; }
```

### Property Value

[bool](#)

# SourcePath

```
public string SourcePath { get; set; }
```

Property Value

[string](#)

## Methods

Process<T>(IObservable<T>)

```
public IObservable<T> Process<T>(IObservable<T> source)
```

Parameters

source [IObservable](#)<T>

Returns

[IObservable](#)<T>

Type Parameters

T

# Class FramestampSoftwareEvent

Namespace: [AllenNeuralDynamics.Core](#)

Assembly: AllenNeuralDynamics.Core.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Transform)]
public class FramestampSoftwareEvent
```

## Inheritance

[object](#) ← FramestampSoftwareEvent

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Methods

Process(IObservable<Tuple<SoftwareEvent,  
RenderSynchState>>)

```
public IObservable<SoftwareEvent> Process(IObservable<Tuple<SoftwareEvent,
RenderSynchState>> source)
```

## Parameters

source [IObservable](#)<[Tuple](#)<SoftwareEvent, RenderSynchState>>

## Returns

[IObservable](#)<SoftwareEvent>

Process(IObservable<Tuple<SoftwareEvent, int>>)

```
public IObservable<SoftwareEvent> Process(IObservable<Tuple<SoftwareEvent, int>> source)
```

Parameters

source [IObservable<Tuple<SoftwareEvent, int>>](#)

Returns

[IObservable<SoftwareEvent>](#)

# Class JsonWriter

Namespace: [AllenNeuralDynamics.Core](#)

Assembly: AllenNeuralDynamics.Core.dll

```
public class JsonWriter : StreamSink<string, StreamWriter>
```

## Inheritance

[object](#) ← [StreamSink](#) ← [StreamSink<string, StreamWriter>](#) ← [JsonWriter](#)

## Inherited Members

[StreamSink<string, StreamWriter>.CreateWriter\(Stream\)](#) ,  
[StreamSink<string, StreamWriter>.Write\(StreamWriter, string\)](#) ,  
[StreamSink<string, StreamWriter>.Process<TElement>\(IObservable<TElement>, Func<TElement, string>\)](#) ,  
[StreamSink<string, StreamWriter>.Process<TElement>\(IObservable<TElement>, Func<TElement, string>, string\)](#) ,  
[StreamSink<string, StreamWriter>.Process\(IObservable<string>\)](#) , [StreamSink.Path](#) ,  
[StreamSink.Suffix](#) , [StreamSink.Overwrite](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### CreateWriter(Stream)

When overridden in a derived class, creates the object that will be responsible for writing the input elements to the specified [Stream](#).

```
protected override StreamWriter CreateWriter(Stream stream)
```

#### Parameters

**stream** [Stream](#)

The stream on which the elements should be written.

## Returns

### [StreamWriter](#)

The object that will be used to write elements into the stream.

## Write(StreamWriter, string)

When overridden in a derived class, writes a new element using the specified writer.

```
protected override void Write(StreamWriter writer, string input)
```

### Parameters

#### writer [StreamWriter](#)

The writer used to push elements into the stream.

#### input [string](#)

The input element that should be pushed into the stream.

# Enum Modality

Namespace: [AllenNeuralDynamics.Core](#)

Assembly: AllenNeuralDynamics.Core.dll

```
public enum Modality
```

## Fields

Behavior = 1

BehaviorVideos = 4

Confocal = 2

Ecephys = 3

Electromyography = 5

Fib = 6

Fmost = 7

Icephys = 8

Isi = 9

Merfish = 10

Mri = 11

P0phys = 12

Slap = 13

Spim = 14

# Class StartProcessOnNewConsole

Namespace: [AllenNeuralDynamics.Core](#)

Assembly: AllenNeuralDynamics.Core.dll

```
public class StartProcessOnNewConsole : Source<Unit>
```

## Inheritance

[object](#) ← [Source](#) <[Unit](#)> ← StartProcessOnNewConsole

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Properties

### Arguments

```
public string Arguments { get; set; }
```

#### Property Value

[string](#)

### FileName

```
[FileNameFilter("Executable files|*.exe|All Files|*.*")]
public string FileName { get; set; }
```

#### Property Value

[string](#)

# Methods

## Generate()

Generates an observable sequence of data elements.

```
public override IObservable<Unit> Generate()
```

Returns

[IObservable](#)<[Unit](#)>

An observable sequence of data elements of type [Unit](#).

## Generate<TSource>(IObservable<TSource>)

```
public IObservable<Unit> Generate<TSource>(IObservable<TSource> source)
```

Parameters

source [IObservable](#)<TSource>

Returns

[IObservable](#)<[Unit](#)>

Type Parameters

TSource

# Class StartRobocopy

Namespace: [AllenNeuralDynamics.Core](#)

Assembly: AllenNeuralDynamics.Core.dll

```
public class StartRobocopy : Source<Unit>
```

## Inheritance

[object](#) ← [Source](#) <[Unit](#)> ← StartRobocopy

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Properties

### Arguments

```
public string Arguments { get; set; }
```

#### Property Value

[string](#)

### CreateLog

```
public bool CreateLog { get; set; }
```

#### Property Value

[bool](#)

### Destination

```
public string Destination { get; set; }
```

Property Value

[string](#)

## ProcessWindowStyle

```
public ProcessWindowStyle ProcessWindowStyle { get; set; }
```

Property Value

[ProcessWindowStyle](#)

## Source

```
public string Source { get; set; }
```

Property Value

[string](#)

## Methods

### Generate()

Generates an observable sequence of data elements.

```
public override IObservable<Unit> Generate()
```

Returns

[IObservable](#)<[Unit](#)>

An observable sequence of data elements of type [Unit](#).



# Class StripDirectoryPath

Namespace: [AllenNeuralDynamics.Core](#)

Assembly: AllenNeuralDynamics.Core.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Transform)]
public class StripDirectoryPath
```

## Inheritance

[object](#) ← StripDirectoryPath

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Methods

### Process(I.observable<string>)

```
public I.observable<string> Process(I.observable<string> source)
```

#### Parameters

source [I.observable](#)<[string](#)>

#### Returns

[I.observable](#)<[string](#)>

# Class ThrowException

Namespace: [AllenNeuralDynamics.Core](#)

Assembly: AllenNeuralDynamics.Core.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Combinator)]
public class ThrowException
```

## Inheritance

[object](#) ← ThrowException

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Message

```
public string Message { get; set; }
```

## Property Value

[string](#)

# Methods

## Process<TSource>(IObservable<TSource>)

```
public IObservable<TSource> Process<TSource>(IObservable<TSource> source)
```

## Parameters

**source** [IObservable](#)<TSource>

Returns

[IObservable](#)<TSource>

Type Parameters

TSource

# Class TimeSpanFromSeconds

Namespace: [AllenNeuralDynamics.Core](#)

Assembly: AllenNeuralDynamics.Core.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Transform)]
public class TimeSpanFromSeconds
```

## Inheritance

[object](#) ← TimeSpanFromSeconds

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Methods

### Process(I Observable<double>)

```
public I Observable<TimeSpan> Process(I Observable<double> source)
```

#### Parameters

source [I Observable](#)<[double](#)>

#### Returns

[I Observable](#)<[TimeSpan](#)>

# Class TimestampSoftwareEvent

Namespace: [AllenNeuralDynamics.Core](#)

Assembly: AllenNeuralDynamics.Core.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Transform)]
public class TimestampSoftwareEvent
```

## Inheritance

[object](#) ← TimestampSoftwareEvent

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Methods

### Process(I Observable<Timestamped<SoftwareEvent>>)

```
public I Observable<SoftwareEvent> Process(I Observable<Timestamped<SoftwareEvent>> source)
```

#### Parameters

source [I Observable](#)<[Timestamped](#)<[SoftwareEvent](#)>>

#### Returns

[I Observable](#)<[SoftwareEvent](#)>

### Process(I Observable< Tuple<SoftwareEvent, HarpMessage>>)

```
public I Observable<SoftwareEvent> Process(I Observable< Tuple<SoftwareEvent,
HarpMessage>> source)
```

Parameters

source [IObservable<Tuple<SoftwareEvent, HarpMessage>>](#)

Returns

[IObservable<SoftwareEvent>](#)

Process([IObservable<Tuple<SoftwareEvent, double>>](#))

```
public IObservable<SoftwareEvent> Process(IObservable<Tuple<SoftwareEvent, double>> source)
```

Parameters

source [IObservable<Tuple<SoftwareEvent, double>>](#)

Returns

[IObservable<SoftwareEvent>](#)

# Enum Verbosity

Namespace: [AllenNeuralDynamics.Core](#)

Assembly: AllenNeuralDynamics.Core.dll

```
public enum Verbosity
```

## Fields

Debug = 7

Error = 3

Fatal = 2

Info = 5

Panic = 1

Quiet = 0

Trace = 8

Verbose = 6

Warning = 4

# Class YamlWriter

Namespace: [AllenNeuralDynamics.Core](#)

Assembly: AllenNeuralDynamics.Core.dll

```
public class YamlWriter : StreamSink<string, StreamWriter>
```

## Inheritance

[object](#) ← [StreamSink](#) ← [StreamSink<string, StreamWriter>](#) ← [YamlWriter](#)

## Inherited Members

[StreamSink<string, StreamWriter>.CreateWriter\(Stream\)](#) ,  
[StreamSink<string, StreamWriter>.Write\(StreamWriter, string\)](#) ,  
[StreamSink<string, StreamWriter>.Process<TElement>\(IObservable<TElement>, Func<TElement, string>\)](#) ,  
[StreamSink<string, StreamWriter>.Process<TElement>\(IObservable<TElement>, Func<TElement, string>, string\)](#) ,  
[StreamSink<string, StreamWriter>.Process\(IObservable<string>\)](#) , [StreamSink.Path](#) ,  
[StreamSink.Suffix](#) , [StreamSink.Overwrite](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### CreateWriter(Stream)

When overridden in a derived class, creates the object that will be responsible for writing the input elements to the specified [Stream](#).

```
protected override StreamWriter CreateWriter(Stream stream)
```

#### Parameters

**stream** [Stream](#)

The stream on which the elements should be written.

## Returns

### [StreamWriter](#)

The object that will be used to write elements into the stream.

## Write(StreamWriter, string)

When overridden in a derived class, writes a new element using the specified writer.

```
protected override void Write(StreamWriter writer, string input)
```

### Parameters

#### writer [StreamWriter](#)

The writer used to push elements into the stream.

#### input [string](#)

The input element that should be pushed into the stream.

# Namespace AllenNeuralDynamics.Core.Design Classes

[AccumulateToImmutableList](#)

[AddImageWeighted](#)

[AnnotationControl](#)

[AnnotationSource](#)

[AnnotationSourceVisualizer](#)

[Annotation<TMetadata>](#)

[IplImageRotateVisualizer](#)

[IplImageSaturationVisualizer](#)

[MessageBox](#)

[PropertyGridVisualizer](#)

[PushButtonControl](#)

[PushButtonVisualizer](#)

[SoftwareEventVisualizer](#)

[ToggleButtonStateControl](#)

[ToggleButtonVisualizer](#)

[ToggleEnabledEventArgs](#)

# Class AccumulateToImmutableList

Namespace: [AllenNeuralDynamics.Core.Design](#)

Assembly: AllenNeuralDynamics.Core.Design.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Transform)]
public class AccumulateToImmutableList
```

## Inheritance

[object](#) ← AccumulateToImmutableList

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Methods

### Process<T>(IObservable<T>)

```
public IObservable<ImmutableList<T>> Process<T>(IObservable<T> source)
```

#### Parameters

source [IObservable](#)<T>

#### Returns

[IObservable](#)<[ImmutableList](#)<T>>

#### Type Parameters

T

# Class AddImageWeighted

Namespace: [AllenNeuralDynamics.Core.Design](#)

Assembly: AllenNeuralDynamics.Core.Design.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Transform)]
public class AddImageWeighted
```

## Inheritance

[object](#) ← AddImageWeighted

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Alpha

```
[Range(0, 1)]
public float Alpha { get; set; }
```

## Property Value

[float](#)

# Methods

## Process(IObservable<Tuple<IpIImage, IpIImage>>)

```
public IObservable<IpIImage> Process(IObservable<Tuple<IpIImage, IpIImage>> source)
```

## Parameters

source [IObservable](#)<[Tuple](#)<[IplImage](#), [IplImage](#)>>

Returns

[IObservable](#)<[IplImage](#)>

# Class AnnotationControl

Namespace: [AllenNeuralDynamics.Core.Design](#)

Assembly: AllenNeuralDynamics.Core.Design.dll

```
public class AnnotationControl : UserControl, IDropTarget, ISynchronizeInvoke, IWin32Window,  
IBindableComponent, IComponent, IDisposable, IContainerControl
```

## Inheritance

[object](#) ← [MarshalByRefObject](#) ← [Component](#) ← [Control](#) ← [ScrollableControl](#) ←  
[ContainerControl](#) ← [UserControl](#) ← AnnotationControl

## Implements

[IDropTarget](#), [ISynchronizeInvoke](#), [IWin32Window](#), [IBindableComponent](#), [IComponent](#),  
[IDisposable](#), [IContainerControl](#)

## Inherited Members

[UserControl.ValidateChildren\(\)](#), [UserControl.ValidateChildren\(ValidationConstraints\)](#),  
[UserControl.OnCreateControl\(\)](#), [UserControl.OnLoad\(EventArgs\)](#),  
[UserControl.OnResize\(EventArgs\)](#), [UserControl.OnMouseDown\(MouseEventArgs\)](#),  
[UserControl.WndProc\(ref Message\)](#), [UserControl.AutoSize](#), [UserControl.AutoSizeMode](#),  
[UserControl.AutoValidate](#), [UserControl.BorderStyle](#), [UserControl.CreateParams](#),  
[UserControl.DefaultSize](#), [UserControl.AutoSizeChanged](#), [UserControl.AutoValidateChanged](#),  
[UserControl.Load](#), [ContainerControl.AdjustFormScrollbars\(bool\)](#),  
[ContainerControl.OnAutoValidateChanged\(EventArgs\)](#),  
[ContainerControl.OnFontChanged\(EventArgs\)](#), [ContainerControl.OnLayout\(LayoutEventArgs\)](#),  
[ContainerControl.OnParentChanged\(EventArgs\)](#), [ContainerControl.PerformLayout\(\)](#),  
[ContainerControl.ProcessDialogChar\(char\)](#), [ContainerControl.ProcessDialogKey\(Keys\)](#),  
[ContainerControl.ProcessCmdKey\(ref Message, Keys\)](#), [ContainerControl.ProcessMnemonic\(char\)](#),  
[ContainerControl.ProcessTabKey\(bool\)](#), [ContainerControl.Select\(bool, bool\)](#),  
[ContainerControl.UpdateDefaultButton\(\)](#), [ContainerControl.Validate\(\)](#),  
[ContainerControl.Validate\(bool\)](#), [ContainerControl.AutoScaleDimensions](#),  
[ContainerControl.AutoScaleFactor](#), [ContainerControl.AutoScaleMode](#),  
[ContainerControl.BindingContext](#), [ContainerControl.CanEnableIme](#),  
[ContainerControl.ActiveControl](#), [ContainerControl.CurrentAutoSizeDimensions](#),  
[ContainerControl.ParentForm](#), [ScrollableControl.ScrollStateAutoScrolling](#),  
[ScrollableControl.ScrollStateHScrollVisible](#), [ScrollableControl.ScrollStateVScrollVisible](#),  
[ScrollableControl.ScrollStateUserHasScrolled](#), [ScrollableControl.ScrollStateFullDrag](#),  
[ScrollableControl.GetScrollState\(int\)](#), [ScrollableControl.OnMouseWheel\(MouseEventArgs\)](#),

[ScrollableControl.OnRightToLeftChanged\(EventArgs\)](#) ,  
[ScrollableControl.OnPaintBackground\(PaintEventArgs\)](#) ,  
[ScrollableControl.OnPaddingChanged\(EventArgs\)](#) , [ScrollableControl.OnVisibleChanged\(EventArgs\)](#) ,  
[ScrollableControl.ScaleControl\(SizeF, BoundsSpecified\)](#) ,  
[ScrollableControl.SetDisplayRectLocation\(int, int\)](#) , [ScrollableControl.ScrollControlIntoView\(Control\)](#) ,  
[ScrollableControl.ScrollToControl\(Control\)](#) , [ScrollableControl.OnScroll\(ScrollEventArgs\)](#) ,  
[ScrollableControl.SetAutoScrollMargin\(int, int\)](#) , [ScrollableControl.SetScrollState\(int, bool\)](#) ,  
[ScrollableControl.AutoScroll](#) , [ScrollableControl.AutoScrollMargin](#) ,  
[ScrollableControl.AutoScrollPosition](#) , [ScrollableControl.AutoScrollMinSize](#) ,  
[ScrollableControl.DisplayRectangle](#) , [ScrollableControl.HScroll](#) , [ScrollableControl.HorizontalScroll](#) ,  
[ScrollableControl.VScroll](#) , [ScrollableControl.VerticalScroll](#) , [ScrollableControl.Scroll](#) ,  
[Control.GetAccessibilityObjectById\(int\)](#) , [Control.SetAutoSizeMode\(AutoSizeMode\)](#) ,  
[Control.GetAutoSizeMode\(\)](#) , [Control.GetPreferredSize\(Size\)](#) ,  
[Control.AccessibilityNotifyClients\(AccessibleEvents, int\)](#) ,  
[Control.AccessibilityNotifyClients\(AccessibleEvents, int, int\)](#) , [Control.BeginInvoke\(Delegate\)](#) ,  
[Control.BeginInvoke\(Delegate, params object\[\]\)](#) , [Control.BringToFront\(\)](#) ,  
[Control.Contains\(Control\)](#) , [Control.CreateAccessibilityInstance\(\)](#) , [Control.CreateControlsInstance\(\)](#) ,  
[Control.CreateGraphics\(\)](#) , [Control.CreateHandle\(\)](#) , [Control.CreateControl\(\)](#) ,  
[Control.DefWndProc\(ref Message\)](#) , [Control.DestroyHandle\(\)](#) ,  
[Control.DoDragDrop\(object, DragDropEffects\)](#) , [Control.DrawToBitmap\(Bitmap, Rectangle\)](#) ,  
[Control.EndInvoke\(IAsyncResult\)](#) , [Control.FindForm\(\)](#) , [Control.GetTopLevel\(\)](#) ,  
[Control.RaiseKeyEvent\(object, KeyEventArgs\)](#) , [Control.RaiseMouseEvent\(object, MouseEventArgs\)](#) ,  
[Control.Focus\(\)](#) , [Control.FromChildHandle\(IntPtr\)](#) , [Control.FromHandle\(IntPtr\)](#) ,  
[Control.GetChildAtPoint\(Point, GetChildAtPointSkip\)](#) , [Control.GetChildAtPoint\(Point\)](#) ,  
[Control.GetContainerControl\(\)](#) , [Control.GetScaledBounds\(Rectangle, SizeF, BoundsSpecified\)](#) ,  
[Control.GetNextControl\(Control, bool\)](#) , [Control.GetStyle\(ControlStyles\)](#) , [Control.Hide\(\)](#) ,  
[Control.InitLayout\(\)](#) , [Control.Invalidate\(Region\)](#) , [Control.Invalidate\(Region, bool\)](#) ,  
[Control.Invalidate\(\)](#) , [Control.Invalidate\(bool\)](#) , [Control.Invalidate\(Rectangle\)](#) ,  
[Control.Invalidate\(Rectangle, bool\)](#) , [Control.Invoke\(Delegate\)](#) ,  
[Control.Invoke\(Delegate, params object\[\]\)](#) , [Control.InvokePaint\(Control, PaintEventArgs\)](#) ,  
[Control.InvokePaintBackground\(Control, PaintEventArgs\)](#) , [Control.IsKeyLocked\(Keys\)](#) ,  
[Control.IsInputChar\(char\)](#) , [Control.IsInputKey\(Keys\)](#) , [Control.IsMnemonic\(char, string\)](#) ,  
[Control.LogicalToDeviceUnits\(int\)](#) , [Control.ScaleBitmapLogicalToDevice\(ref Bitmap\)](#) ,  
[Control.NotifyInvalidate\(Rectangle\)](#) , [Control.InvokeOnClick\(Control, EventArgs\)](#) ,  
[Control.OnAutoSizeChanged\(EventArgs\)](#) , [Control.OnBackColorChanged\(EventArgs\)](#) ,  
[Control.OnBackgroundImageChanged\(EventArgs\)](#) ,  
[Control.OnBackgroundImageLayoutChanged\(EventArgs\)](#) ,  
[Control.OnBindingContextChanged\(EventArgs\)](#) , [Control.OnCausesValidationChanged\(EventArgs\)](#) ,  
[Control.OnContextMenuChanged\(EventArgs\)](#) , [Control.OnContextMenuStripChanged\(EventArgs\)](#) ,  
[Control.OnCursorChanged\(EventArgs\)](#) , [Control.OnDockChanged\(EventArgs\)](#) ,

[Control.OnEnabledChanged\(EventArgs\)](#) , [Control.OnForeColorChanged\(EventArgs\)](#) ,  
[Control.OnNotifyMessage\(Message\)](#) , [Control.OnParentBackColorChanged\(EventArgs\)](#) ,  
[Control.OnParentBackgroundImageChanged\(EventArgs\)](#) ,  
[Control.OnParentBindingContextChanged\(EventArgs\)](#) , [Control.OnParentCursorChanged\(EventArgs\)](#) ,  
[Control.OnParentEnabledChanged\(EventArgs\)](#) , [Control.OnParentFontChanged\(EventArgs\)](#) ,  
[Control.OnParentForeColorChanged\(EventArgs\)](#) , [Control.OnParentRightToLeftChanged\(EventArgs\)](#) ,  
[Control.OnParentVisibleChanged\(EventArgs\)](#) , [Control.OnPrint\(PaintEventArgs\)](#) ,  
[Control.OnTabIndexChanged\(EventArgs\)](#) , [Control.OnTabStopChanged\(EventArgs\)](#) ,  
[Control.OnTextChanged\(EventArgs\)](#) , [Control.OnClick\(EventArgs\)](#) ,  
[Control.OnClientSizeChanged\(EventArgs\)](#) , [Control.OnControlAdded\(ControlEventArgs\)](#) ,  
[Control.OnControlRemoved\(ControlEventArgs\)](#) , [Control.OnHandleCreated\(EventArgs\)](#) ,  
[Control.OnLocationChanged\(EventArgs\)](#) , [Control.OnHandleDestroyed\(EventArgs\)](#) ,  
[Control.OnDoubleClick\(EventArgs\)](#) , [Control.OnDragEnter\(DragEventArgs\)](#) ,  
[Control.OnDragOver\(DragEventArgs\)](#) , [Control.OnDragLeave\(EventArgs\)](#) ,  
[Control.OnDragDrop\(DragEventArgs\)](#) , [Control.OnGiveFeedback\(GiveFeedbackEventArgs\)](#) ,  
[Control.OnEnter\(EventArgs\)](#) , [Control.InvokeGotFocus\(Control, EventArgs\)](#) ,  
[Control.OnGotFocus\(EventArgs\)](#) , [Control.OnHelpRequested\(HelpEventArgs\)](#) ,  
[Control.OnInvalidated\(InvalidEventArgs\)](#) , [Control.OnKeyDown\(KeyEventEventArgs\)](#) ,  
[Control.OnKeyPress\(KeyEventEventArgs\)](#) , [Control.OnKeyUp\(KeyEventEventArgs\)](#) ,  
[Control.OnLeave\(EventArgs\)](#) , [Control.InvokeLostFocus\(Control, EventArgs\)](#) ,  
[Control.OnLostFocus\(EventArgs\)](#) , [Control.OnMarginChanged\(EventArgs\)](#) ,  
[Control.OnMouseDoubleClick\(MouseEventArgs\)](#) , [Control.OnMouseClick\(MouseEventArgs\)](#) ,  
[Control.OnMouseCaptureChanged\(EventArgs\)](#) , [Control.OnMouseEnter\(EventArgs\)](#) ,  
[Control.OnMouseLeave\(EventArgs\)](#) , [Control.OnDpiChangedBeforeParent\(EventArgs\)](#) ,  
[Control.OnDpiChangedAfterParent\(EventArgs\)](#) , [Control.OnMouseHover\(EventArgs\)](#) ,  
[Control.OnMouseMove\(MouseEventArgs\)](#) , [Control.OnMouseUp\(MouseEventArgs\)](#) ,  
[Control.OnMove\(EventArgs\)](#) , [Control.OnPaint\(PaintEventArgs\)](#) ,  
[Control.OnQueryContinueDrag\(QueryContinueDragEventArgs\)](#) ,  
[Control.OnRegionChanged\(EventArgs\)](#) , [Control.OnPreviewKeyDown\(PreviewKeyDownEventArgs\)](#) ,  
[Control.OnSizeChanged\(EventArgs\)](#) , [Control.OnChangeUICues\(UICuesEventArgs\)](#) ,  
[Control.OnStyleChanged\(EventArgs\)](#) , [Control.OnSystemColorsChanged\(EventArgs\)](#) ,  
[Control.OnValidating\(CancelEventArgs\)](#) , [Control.OnValidated\(EventArgs\)](#) ,  
[Control.RescaleConstantsForDpi\(int, int\)](#) , [Control.PerformLayout\(\)](#) ,  
[Control.PerformLayout\(Control, string\)](#) , [Control.PointToClient\(Point\)](#) , [Control.PointToScreen\(Point\)](#) ,  
[Control.PreProcessMessage\(ref Message\)](#) , [Control.PreProcessControlMessage\(ref Message\)](#) ,  
[Control.ProcessKeyEventArgs\(ref Message\)](#) , [Control.ProcessKeyMessage\(ref Message\)](#) ,  
[Control.ProcessKeyPreview\(ref Message\)](#) , [Control.RaiseDragEvent\(object, DragEventArgs\)](#) ,  
[Control.RaisePaintEvent\(object, PaintEventArgs\)](#) , [Control.RecreateHandle\(\)](#) ,  
[Control.RectangleToClient\(Rectangle\)](#) , [Control.RectangleToScreen\(Rectangle\)](#) ,  
[Control.ReflectMessage\(IntPtr, ref Message\)](#) , [Control.Refresh\(\)](#) , [Control.ResetMouseEventArgs\(\)](#) ,

[Control.ResetText\(\)](#) , [Control.ResumeLayout\(\)](#) , [Control.ResumeLayout\(bool\)](#) , [Control.Scale\(SizeF\)](#) ,  
[Control.Select\(\)](#) , [Control.SelectNextControl\(Control, bool, bool, bool, bool\)](#) , [Control.SendToBack\(\)](#) ,  
[Control.SetBounds\(int, int, int, int\)](#) , [Control.SetBounds\(int, int, int, int, BoundsSpecified\)](#) ,  
[Control.SetBoundsCore\(int, int, int, int, BoundsSpecified\)](#) , [Control.SetClientSizeCore\(int, int\)](#) ,  
[Control.SizeFromClientSize\(Size\)](#) , [Control.SetStyle\(ControlStyles, bool\)](#) , [Control.SetTopLevel\(bool\)](#) ,  
[Control.SetVisibleCore\(bool\)](#) , [Control.RtlTranslateAlignment\(HorizontalAlignment\)](#) ,  
[Control.RtlTranslateAlignment\(LeftRightAlignment\)](#) ,  
[Control.RtlTranslateAlignment\(ContentAlignment\)](#) ,  
[Control.RtlTranslateHorizontal\(HorizontalAlignment\)](#) ,  
[Control.RtlTranslateLeftRight\(LeftRightAlignment\)](#) , [Control.RtlTranslateContent\(ContentAlignment\)](#) ,  
[Control.Show\(\)](#) , [Control.SuspendLayout\(\)](#) , [Control.Update\(\)](#) , [Control.UpdateBounds\(\)](#) ,  
[Control.UpdateBounds\(int, int, int, int\)](#) , [Control.UpdateBounds\(int, int, int, int, int, int\)](#) ,  
[Control.UpdateZOrder\(\)](#) , [Control.UpdateStyles\(\)](#) , [Control.OnImeModeChanged\(EventArgs\)](#) ,  
[Control.AccessibilityObject](#) , [Control.AccessibleDefaultActionDescription](#) ,  
[Control.AccessibleDescription](#) , [Control.AccessibleName](#) , [Control.AccessibleRole](#) ,  
[Control.AllowDrop](#) , [Control.Anchor](#) , [Control.AutoScrollOffset](#) , [Control.LayoutEngine](#) ,  
[Control.BackColor](#) , [Control.BackgroundImage](#) , [Control.BackgroundImageLayout](#) ,  
[Control.Bottom](#) , [Control.Bounds](#) , [Control.CanFocus](#) , [Control.CanRaiseEvents](#) ,  
[Control.CanSelect](#) , [Control.Capture](#) , [Control.CausesValidation](#) ,  
[Control.CheckForIllegalCrossThreadCalls](#) , [Control.ClientRectangle](#) , [Control.ClientSize](#) ,  
[Control.CompanyName](#) , [Control.ContainsFocus](#) , [Control.ContextMenu](#) ,  
[Control.ContextMenuStrip](#) , [Control.Controls](#) , [Control.Created](#) , [Control.Cursor](#) ,  
[Control.DataBindings](#) , [Control.DefaultBackColor](#) , [Control.DefaultCursor](#) , [Control.DefaultFont](#) ,  
[Control.DefaultForeColor](#) , [Control.DefaultMargin](#) , [Control.DefaultMaximumSize](#) ,  
[Control.DefaultMinimumSize](#) , [Control.DefaultPadding](#) , [Control.DeviceDpi](#) , [Control.IsDisposed](#) ,  
[Control.Disposing](#) , [Control.Dock](#) , [Control.DoubleBuffered](#) , [Control.Enabled](#) , [Control.Focused](#) ,  
[Control.Font](#) , [Control.FontHeight](#) , [Control.ForeColor](#) , [Control.Handle](#) , [Control.HasChildren](#) ,  
[Control.Height](#) , [Control.IsHandleCreated](#) , [Control.InvokeRequired](#) , [Control.IsAccessible](#) ,  
[Control.IsMirrored](#) , [Control.Left](#) , [Control.Location](#) , [Control.Margin](#) , [Control.MaximumSize](#) ,  
[Control.MinimumSize](#) , [Control.ModifierKeys](#) , [Control.MouseButtons](#) , [Control.mousePosition](#) ,  
[Control.Name](#) , [Control.Parent](#) , [Control.ProductName](#) , [Control.ProductVersion](#) ,  
[Control.RecreatingHandle](#) , [Control.Region](#) , [Control.ResizeRedraw](#) , [Control.Right](#) ,  
[Control.RightToLeft](#) , [Control.ScaleChildren](#) , [Control.Site](#) , [Control.Size](#) , [Control.TabIndex](#) ,  
[Control.TabStop](#) , [Control.Tag](#) , [Control.Text](#) , [Control.Top](#) , [Control.TopLevelControl](#) ,  
[Control>ShowKeyboardCues](#) , [Control>ShowFocusCues](#) , [Control.UseWaitCursor](#) , [Control.Visible](#) ,  
[Control.Width](#) , [Control.PreferredSize](#) , [Control.Padding](#) , [Control.DefaultImeMode](#) ,  
[Control.ImeMode](#) , [Control.ImeModeBase](#) , [Control.PropagatingImeMode](#) ,  
[Control.BackColorChanged](#) , [Control.BackgroundImageChanged](#) ,  
[Control.BackgroundImageLayoutChanged](#) , [Control.BindingContextChanged](#) ,  
[Control.CausesValidationChanged](#) , [Control.ClientSizeChanged](#) , [Control.ContextMenuChanged](#) ,

[Control.ContextMenuStripChanged](#) , [Control.CursorChanged](#) , [Control.DockChanged](#) ,  
[Control.EnabledChanged](#) , [Control.FontChanged](#) , [Control.ForeColorChanged](#) ,  
[Control.LocationChanged](#) , [Control.MarginChanged](#) , [Control.RegionChanged](#) ,  
[Control.RightToLeftChanged](#) , [Control.SizeChanged](#) , [Control.TabIndexChanged](#) ,  
[Control.TabStopChanged](#) , [Control.TextChanged](#) , [Control.VisibleChanged](#) , [Control.Click](#) ,  
[Control.ControlAdded](#) , [Control.ControlRemoved](#) , [Control.DragDrop](#) , [Control.DragEnter](#) ,  
[Control.DragOver](#) , [Control.DragLeave](#) , [Control.GiveFeedback](#) , [Control.HandleCreated](#) ,  
[Control.HandleDestroyed](#) , [Control.HelpRequested](#) , [Control.Invalidate](#) ,  
[Control.PaddingChanged](#) , [Control.Paint](#) , [Control.QueryContinueDrag](#) ,  
[Control.QueryAccessibilityHelp](#) , [Control.DoubleClick](#) , [Control.Enter](#) , [Control.GotFocus](#) ,  
[Control.KeyDown](#) , [Control.KeyPress](#) , [Control.KeyUp](#) , [Control.Layout](#) , [Control.Leave](#) ,  
[Control.LostFocus](#) , [Control.MouseClick](#) , [Control.MouseDoubleClick](#) ,  
[Control.MouseCaptureChanged](#) , [Control.MouseDown](#) , [Control.MouseEnter](#) ,  
[Control.MouseLeave](#) , [Control.DpiChangedBeforeParent](#) , [Control.DpiChangedAfterParent](#) ,  
[Control.MouseHover](#) , [Control.MouseMove](#) , [Control.MouseUp](#) , [Control.MouseWheel](#) ,  
[Control.Move](#) , [Control.PreviewKeyDown](#) , [Control.Resize](#) , [Control.ChangeUICues](#) ,  
[Control.StyleChanged](#) , [Control.SystemColorsChanged](#) , [Control.Validating](#) , [Control.Validated](#) ,  
[Control.ParentChanged](#) , [Control.ImeModeChanged](#) , [Component.Dispose\(\)](#) ,  
[Component.GetService\(Type\)](#) , [Component.ToString\(\)](#) , [Component.Events](#) ,  
[Component.Container](#) , [Component.DesignMode](#) , [Component.Dispose](#) ,  
[MarshalByRefObject.MemberwiseClone\(bool\)](#) , [MarshalByRefObject.GetLifetimeService\(\)](#) ,  
[MarshalByRefObject.InitializeLifetimeService\(\)](#) , [MarshalByRefObject.CreateObjRef\(Type\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### AnnotationControl(AnnotationSource)

```
public AnnotationControl(AnnotationSource source)
```

#### Parameters

source [AnnotationSource](#)

## Properties

## Source

```
public AnnotationSource Source { get; }
```

### Property Value

[AnnotationSource](#)

## Methods

### Dispose(bool)

Clean up any resources being used.

```
protected override void Dispose(bool disposing)
```

### Parameters

`disposing` [bool](#)

true if managed resources should be disposed; otherwise, false.

# Class AnnotationSource

Namespace: [AllenNeuralDynamics.Core.Design](#)

Assembly: AllenNeuralDynamics.Core.Design.dll

```
public class AnnotationSource : Annotation<string>
```

## Inheritance

[object](#) ← [Annotation<string>](#) < AnnotationSource

## Inherited Members

[Annotation<string>.OnNext\(string\)](#) , [Annotation<string>.Process\(\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

# Class AnnotationSourceVisualizer

Namespace: [AllenNeuralDynamics.Core.Design](#)

Assembly: AllenNeuralDynamics.Core.Design.dll

```
public class AnnotationSourceVisualizer : DialogTypeVisualizer
```

## Inheritance

[object](#) ← [DialogTypeVisualizer](#) ← AnnotationSourceVisualizer

## Inherited Members

[DialogTypeVisualizer.Visualize\(IObservable<IObservable<object>>, IServiceProvider\)](#) ,  
[DialogTypeVisualizer.SequenceCompleted\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### Load(IServiceProvider)

Loads type visualizer resources using the specified service provider.

```
public override void Load(IServiceProvider provider)
```

#### Parameters

**provider** [IServiceProvider](#)

A service provider object which can be used to obtain visualization, runtime inspection, or other editing services.

### Show(object)

Updates the type visualizer to display the specified value object.

```
public override void Show(object value)
```

## Parameters

**value** [object](#)

The value to visualize.

## Unload()

Unloads all type visualizer resources.

```
public override void Unload()
```

# Class Annotation<TMetadata>

Namespace: [AllenNeuralDynamics.Core.Design](#)

Assembly: AllenNeuralDynamics.Core.Design.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Source)]
[TypeVisualizer(typeof(AnnotationSourceVisualizer))]
public abstract class Annotation<TMetadata>
```

## Type Parameters

**TMetadata**

## Inheritance

[object](#) ← Annotation<TMetadata>

## Derived

[AnnotationSource](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Methods

### OnNext(TMetadata)

```
public void OnNext(TMetadata value)
```

## Parameters

**value** TMetadata

### Process()

```
public virtual IObservable<TMetadata> Process()
```

Returns

[IObservable](#)<TMetadata>

# Class IpllImageRotateVisualizer

Namespace: [AllenNeuralDynamics.Core.Design](#)

Assembly: AllenNeuralDynamics.Core.Design.dll

```
public class IpllImageRotateVisualizer : IpllImageVisualizer
```

## Inheritance

[object](#) ← [DialogTypeVisualizer](#) ← [MashupVisualizer](#) ← [DialogMashupVisualizer](#) ←  
[ImageMashupVisualizer](#) ← [IpllImageVisualizer](#) ← [IpllImageRotateVisualizer](#)

## Inherited Members

[IpllImageVisualizer.ShowMashup\(IList<object>\)](#) , [IpllImageVisualizer.Show\(object\)](#) ,  
[IpllImageVisualizer.Unload\(\)](#) , [IpllImageVisualizer.StatusStripEnabled](#) , [IpllImageVisualizer.StatusStrip](#) ,  
[IpllImageVisualizer.VisualizerCanvas](#) , [ImageMashupVisualizer.UpdateValues\(IList<object>\)](#) ,  
[ImageMashupVisualizer.Visualize\(IObservable<IObservable<object>>, IServiceProvider\)](#) ,  
[ImageMashupVisualizer.VisualizerImage](#) , [MashupVisualizer.LoadMashups\(IServiceProvider\)](#) ,  
[MashupVisualizer.UnloadMashups\(\)](#) , [MashupVisualizer.GetMashupSource\(int, int\)](#) ,  
[MashupVisualizer.MashupSources](#) , [DialogTypeVisualizer.SequenceCompleted\(\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Properties

### InvertHorizontal

```
public bool InvertHorizontal { get; set; }
```

#### Property Value

[bool](#)

### InvertVertical

```
public bool InvertVertical { get; set; }
```

Property Value

[bool](#)

RotateAngle

```
public float RotateAngle { get; set; }
```

Property Value

[float](#)

## Methods

Load(IServiceProvider)

Loads type visualizer resources using the specified service provider.

```
public override void Load(IServiceProvider provider)
```

Parameters

provider [IServiceProvider](#)

A service provider object which can be used to obtain visualization, runtime inspection, or other editing services.

RenderFrame()

Renders all graphics to the visualizer canvas. Override this method to overlay additional graphics elements on top of the image texture.

```
protected override void RenderFrame()
```

# Class IplImageSaturationVisualizer

Namespace: [AllenNeuralDynamics.Core.Design](#)

Assembly: AllenNeuralDynamics.Core.Design.dll

```
public class IplImageSaturationVisualizer : IplImageVisualizer
```

## Inheritance

[object](#) ← [DialogTypeVisualizer](#) ← [MashupVisualizer](#) ← [DialogMashupVisualizer](#) ←  
[ImageMashupVisualizer](#) ← [IplImageVisualizer](#) ← [IplImageSaturationVisualizer](#)

## Inherited Members

[IplImageVisualizer.ShowMashup\(IList<object>\)](#) , [IplImageVisualizer.Show\(object\)](#) ,  
[IplImageVisualizer.Unload\(\)](#) , [IplImageVisualizer.StatusStripEnabled](#) , [IplImageVisualizer.StatusStrip](#) ,  
[IplImageVisualizer.VisualizerCanvas](#) , [ImageMashupVisualizer.UpdateValues\(IList<object>\)](#) ,  
[ImageMashupVisualizer.Visualize\(IObservable<IObservable<object>>, IServiceProvider\)](#) ,  
[ImageMashupVisualizer.VisualizerImage](#) , [MashupVisualizer.LoadMashups\(IServiceProvider\)](#) ,  
[MashupVisualizer.UnloadMashups\(\)](#) , [MashupVisualizer.GetMashupSource\(int, int\)](#) ,  
[MashupVisualizer.MashupSources](#) , [DialogTypeVisualizer.SequenceCompleted\(\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Properties

### maxSaturation

```
public byte maxSaturation { get; set; }
```

#### Property Value

[byte](#)

### minSaturation

```
public byte minSaturation { get; set; }
```

Property Value

[byte](#)

## Methods

### Load(IServiceProvider)

Loads type visualizer resources using the specified service provider.

```
public override void Load(IServiceProvider provider)
```

#### Parameters

[provider](#) [IServiceProvider](#)

A service provider object which can be used to obtain visualization, runtime inspection, or other editing services.

### RenderFrame()

Renders all graphics to the visualizer canvas. Override this method to overlay additional graphics elements on top of the image texture.

```
protected override void RenderFrame()
```

# Class MessageBox

Namespace: [AllenNeuralDynamics.Core.Design](#)

Assembly: AllenNeuralDynamics.Core.Design.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Sink)]
public class MessageBox
```

## Inheritance

[object](#) ← MessageBox

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## MessageBoxIcon

```
public MessageBoxIcon MessageBoxIcon { get; set; }
```

## Property Value

[MessageBoxIcon](#)

## Text

```
public string Text { get; set; }
```

## Property Value

[string](#)

## Title

```
public string Title { get; set; }
```

### Property Value

[string](#)

## Methods

### Process<TSource>(IObservable<TSource>)

```
public IObservable<TSource> Process<TSource>(IObservable<TSource> source)
```

#### Parameters

source [IObservable](#)<TSource>

#### Returns

[IObservable](#)<TSource>

#### Type Parameters

TSource

# Class PropertyGridVisualizer

Namespace: [AllenNeuralDynamics.Core.Design](#)

Assembly: AllenNeuralDynamics.Core.Design.dll

```
public class PropertyGridVisualizer : DialogTypeVisualizer
```

## Inheritance

[object](#) ← [DialogTypeVisualizer](#) ← PropertyGridVisualizer

## Inherited Members

[DialogTypeVisualizer.Visualize\(IObservable<IObservable<object>>, IServiceProvider\)](#) ,  
[DialogTypeVisualizer.SequenceCompleted\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### Load(IServiceProvider)

Loads type visualizer resources using the specified service provider.

```
public override void Load(IServiceProvider provider)
```

#### Parameters

provider [IServiceProvider](#)

A service provider object which can be used to obtain visualization, runtime inspection, or other editing services.

### Show(object)

Updates the type visualizer to display the specified value object.

```
public override void Show(object value)
```

## Parameters

**value** [object](#)

The value to visualize.

## Unload()

Unloads all type visualizer resources.

```
public override void Unload()
```

# Class PushButtonControl

Namespace: [AllenNeuralDynamics.Core.Design](#)

Assembly: AllenNeuralDynamics.Core.Design.dll

```
public class PushButtonControl : UserControl, IDropTarget, ISynchronizeInvoke, IWin32Window,  
IBindableComponent, IComponent, IDisposable, IContainerControl
```

## Inheritance

[object](#) ← [MarshalByRefObject](#) ← [Component](#) ← [Control](#) ← [ScrollableControl](#) ←  
[ContainerControl](#) ← [UserControl](#) ← PushButtonControl

## Implements

[IDropTarget](#), [ISynchronizeInvoke](#), [IWin32Window](#), [IBindableComponent](#), [IComponent](#),  
[IDisposable](#), [IContainerControl](#)

## Inherited Members

[UserControl.ValidateChildren\(\)](#), [UserControl.ValidateChildren\(ValidationConstraints\)](#),  
[UserControl.OnCreateControl\(\)](#), [UserControl.OnLoad\(EventArgs\)](#),  
[UserControl.OnResize\(EventArgs\)](#), [UserControl.OnMouseDown\(MouseEventArgs\)](#),  
[UserControl.WndProc\(ref Message\)](#), [UserControl.AutoSize](#), [UserControl.AutoSizeMode](#),  
[UserControl.AutoValidate](#), [UserControl.BorderStyle](#), [UserControl.CreateParams](#),  
[UserControl.DefaultSize](#), [UserControl.AutoSizeChanged](#), [UserControl.AutoValidateChanged](#),  
[UserControl.Load](#), [ContainerControl.AdjustFormScrollbars\(bool\)](#),  
[ContainerControl.OnAutoValidateChanged\(EventArgs\)](#),  
[ContainerControl.OnFontChanged\(EventArgs\)](#), [ContainerControl.OnLayout\(LayoutEventArgs\)](#),  
[ContainerControl.OnParentChanged\(EventArgs\)](#), [ContainerControl.PerformLayout\(\)](#),  
[ContainerControl.ProcessDialogChar\(char\)](#), [ContainerControl.ProcessDialogKey\(Keys\)](#),  
[ContainerControl.ProcessCmdKey\(ref Message, Keys\)](#), [ContainerControl.ProcessMnemonic\(char\)](#),  
[ContainerControl.ProcessTabKey\(bool\)](#), [ContainerControl.Select\(bool, bool\)](#),  
[ContainerControl.UpdateDefaultButton\(\)](#), [ContainerControl.Validate\(\)](#),  
[ContainerControl.Validate\(bool\)](#), [ContainerControl.AutoScaleDimensions](#),  
[ContainerControl.AutoScaleFactor](#), [ContainerControl.AutoScaleMode](#),  
[ContainerControl.BindingContext](#), [ContainerControl.CanEnableIme](#),  
[ContainerControl.ActiveControl](#), [ContainerControl.CurrentAutoSizeDimensions](#),  
[ContainerControl.ParentForm](#), [ScrollableControl.ScrollStateAutoScrolling](#),  
[ScrollableControl.ScrollStateHScrollVisible](#), [ScrollableControl.ScrollStateVScrollVisible](#),  
[ScrollableControl.ScrollStateUserHasScrolled](#), [ScrollableControl.ScrollStateFullDrag](#),  
[ScrollableControl.GetScrollState\(int\)](#), [ScrollableControl.OnMouseWheel\(MouseEventArgs\)](#),

[ScrollableControl.OnRightToLeftChanged\(EventArgs\)](#) ,  
[ScrollableControl.OnPaintBackground\(PaintEventArgs\)](#) ,  
[ScrollableControl.OnPaddingChanged\(EventArgs\)](#) , [ScrollableControl.OnVisibleChanged\(EventArgs\)](#) ,  
[ScrollableControl.ScaleControl\(SizeF, BoundsSpecified\)](#) ,  
[ScrollableControl.SetDisplayRectLocation\(int, int\)](#) , [ScrollableControl.ScrollControlIntoView\(Control\)](#) ,  
[ScrollableControl.ScrollToControl\(Control\)](#) , [ScrollableControl.OnScroll\(ScrollEventArgs\)](#) ,  
[ScrollableControl.SetAutoScrollMargin\(int, int\)](#) , [ScrollableControl.SetScrollState\(int, bool\)](#) ,  
[ScrollableControl.AutoScroll](#) , [ScrollableControl.AutoScrollMargin](#) ,  
[ScrollableControl.AutoScrollPosition](#) , [ScrollableControl.AutoScrollMinSize](#) ,  
[ScrollableControl.DisplayRectangle](#) , [ScrollableControl.HScroll](#) , [ScrollableControl.HorizontalScroll](#) ,  
[ScrollableControl.VScroll](#) , [ScrollableControl.VerticalScroll](#) , [ScrollableControl.Scroll](#) ,  
[Control.GetAccessibilityObjectById\(int\)](#) , [Control.SetAutoSizeMode\(AutoSizeMode\)](#) ,  
[Control.GetAutoSizeMode\(\)](#) , [Control.GetPreferredSize\(Size\)](#) ,  
[Control.AccessibilityNotifyClients\(AccessibleEvents, int\)](#) ,  
[Control.AccessibilityNotifyClients\(AccessibleEvents, int, int\)](#) , [Control.BeginInvoke\(Delegate\)](#) ,  
[Control.BeginInvoke\(Delegate, params object\[\]\)](#) , [Control.BringToFront\(\)](#) ,  
[Control.Contains\(Control\)](#) , [Control.CreateAccessibilityInstance\(\)](#) , [Control.CreateControlsInstance\(\)](#) ,  
[Control.CreateGraphics\(\)](#) , [Control.CreateHandle\(\)](#) , [Control.CreateControl\(\)](#) ,  
[Control.DefWndProc\(ref Message\)](#) , [Control.DestroyHandle\(\)](#) ,  
[Control.DoDragDrop\(object, DragDropEffects\)](#) , [Control.DrawToBitmap\(Bitmap, Rectangle\)](#) ,  
[Control.EndInvoke\(IAsyncResult\)](#) , [Control.FindForm\(\)](#) , [Control.GetTopLevel\(\)](#) ,  
[Control.RaiseKeyEvent\(object, KeyEventArgs\)](#) , [Control.RaiseMouseEvent\(object, MouseEventArgs\)](#) ,  
[Control.Focus\(\)](#) , [Control.FromChildHandle\(IntPtr\)](#) , [Control.FromHandle\(IntPtr\)](#) ,  
[Control.GetChildAtPoint\(Point, GetChildAtPointSkip\)](#) , [Control.GetChildAtPoint\(Point\)](#) ,  
[Control.GetContainerControl\(\)](#) , [Control.GetScaledBounds\(Rectangle, SizeF, BoundsSpecified\)](#) ,  
[Control.GetNextControl\(Control, bool\)](#) , [Control.GetStyle\(ControlStyles\)](#) , [Control.Hide\(\)](#) ,  
[Control.InitLayout\(\)](#) , [Control.Invalidate\(Region\)](#) , [Control.Invalidate\(Region, bool\)](#) ,  
[Control.Invalidate\(\)](#) , [Control.Invalidate\(bool\)](#) , [Control.Invalidate\(Rectangle\)](#) ,  
[Control.Invalidate\(Rectangle, bool\)](#) , [Control.Invoke\(Delegate\)](#) ,  
[Control.Invoke\(Delegate, params object\[\]\)](#) , [Control.InvokePaint\(Control, PaintEventArgs\)](#) ,  
[Control.InvokePaintBackground\(Control, PaintEventArgs\)](#) , [Control.IsKeyLocked\(Keys\)](#) ,  
[Control.IsInputChar\(char\)](#) , [Control.IsInputKey\(Keys\)](#) , [Control.IsMnemonic\(char, string\)](#) ,  
[Control.LogicalToDeviceUnits\(int\)](#) , [Control.ScaleBitmapLogicalToDevice\(ref Bitmap\)](#) ,  
[Control.NotifyInvalidate\(Rectangle\)](#) , [Control.InvokeOnClick\(Control, EventArgs\)](#) ,  
[Control.OnAutoSizeChanged\(EventArgs\)](#) , [Control.OnBackColorChanged\(EventArgs\)](#) ,  
[Control.OnBackgroundImageChanged\(EventArgs\)](#) ,  
[Control.OnBackgroundImageLayoutChanged\(EventArgs\)](#) ,  
[Control.OnBindingContextChanged\(EventArgs\)](#) , [Control.OnCausesValidationChanged\(EventArgs\)](#) ,  
[Control.OnContextMenuChanged\(EventArgs\)](#) , [Control.OnContextMenuStripChanged\(EventArgs\)](#) ,  
[Control.OnCursorChanged\(EventArgs\)](#) , [Control.OnDockChanged\(EventArgs\)](#) ,

[Control.OnEnabledChanged\(EventArgs\)](#) , [Control.OnForeColorChanged\(EventArgs\)](#) ,  
[Control.OnNotifyMessage\(Message\)](#) , [Control.OnParentBackColorChanged\(EventArgs\)](#) ,  
[Control.OnParentBackgroundImageChanged\(EventArgs\)](#) ,  
[Control.OnParentBindingContextChanged\(EventArgs\)](#) , [Control.OnParentCursorChanged\(EventArgs\)](#) ,  
[Control.OnParentEnabledChanged\(EventArgs\)](#) , [Control.OnParentFontChanged\(EventArgs\)](#) ,  
[Control.OnParentForeColorChanged\(EventArgs\)](#) , [Control.OnParentRightToLeftChanged\(EventArgs\)](#) ,  
[Control.OnParentVisibleChanged\(EventArgs\)](#) , [Control.OnPrint\(PaintEventArgs\)](#) ,  
[Control.OnTabIndexChanged\(EventArgs\)](#) , [Control.OnTabStopChanged\(EventArgs\)](#) ,  
[Control.OnTextChanged\(EventArgs\)](#) , [Control.OnClick\(EventArgs\)](#) ,  
[Control.OnClientSizeChanged\(EventArgs\)](#) , [Control.OnControlAdded\(ControlEventArgs\)](#) ,  
[Control.OnControlRemoved\(ControlEventArgs\)](#) , [Control.OnHandleCreated\(EventArgs\)](#) ,  
[Control.OnLocationChanged\(EventArgs\)](#) , [Control.OnHandleDestroyed\(EventArgs\)](#) ,  
[Control.OnDoubleClick\(EventArgs\)](#) , [Control.OnDragEnter\(DragEventArgs\)](#) ,  
[Control.OnDragOver\(DragEventArgs\)](#) , [Control.OnDragLeave\(EventArgs\)](#) ,  
[Control.OnDragDrop\(DragEventArgs\)](#) , [Control.OnGiveFeedback\(GiveFeedbackEventArgs\)](#) ,  
[Control.OnEnter\(EventArgs\)](#) , [Control.InvokeGotFocus\(Control, EventArgs\)](#) ,  
[Control.OnGotFocus\(EventArgs\)](#) , [Control.OnHelpRequested\(HelpEventArgs\)](#) ,  
[Control.OnInvalidated\(InvalidEventArgs\)](#) , [Control.OnKeyDown\(KeyEventEventArgs\)](#) ,  
[Control.OnKeyPress\(KeyEventEventArgs\)](#) , [Control.OnKeyUp\(KeyEventEventArgs\)](#) ,  
[Control.OnLeave\(EventArgs\)](#) , [Control.InvokeLostFocus\(Control, EventArgs\)](#) ,  
[Control.OnLostFocus\(EventArgs\)](#) , [Control.OnMarginChanged\(EventArgs\)](#) ,  
[Control.OnMouseDoubleClick\(MouseEventArgs\)](#) , [Control.OnMouseClick\(MouseEventArgs\)](#) ,  
[Control.OnMouseCaptureChanged\(EventArgs\)](#) , [Control.OnMouseEnter\(EventArgs\)](#) ,  
[Control.OnMouseLeave\(EventArgs\)](#) , [Control.OnDpiChangedBeforeParent\(EventArgs\)](#) ,  
[Control.OnDpiChangedAfterParent\(EventArgs\)](#) , [Control.OnMouseHover\(EventArgs\)](#) ,  
[Control.OnMouseMove\(MouseEventArgs\)](#) , [Control.OnMouseUp\(MouseEventArgs\)](#) ,  
[Control.OnMove\(EventArgs\)](#) , [Control.OnPaint\(PaintEventArgs\)](#) ,  
[Control.OnQueryContinueDrag\(QueryContinueDragEventArgs\)](#) ,  
[Control.OnRegionChanged\(EventArgs\)](#) , [Control.OnPreviewKeyDown\(PreviewKeyDownEventArgs\)](#) ,  
[Control.OnSizeChanged\(EventArgs\)](#) , [Control.OnChangeUICues\(UICuesEventArgs\)](#) ,  
[Control.OnStyleChanged\(EventArgs\)](#) , [Control.OnSystemColorsChanged\(EventArgs\)](#) ,  
[Control.OnValidating\(CancelEventArgs\)](#) , [Control.OnValidated\(EventArgs\)](#) ,  
[Control.RescaleConstantsForDpi\(int, int\)](#) , [Control.PerformLayout\(\)](#) ,  
[Control.PerformLayout\(Control, string\)](#) , [Control.PointToClient\(Point\)](#) , [Control.PointToScreen\(Point\)](#) ,  
[Control.PreProcessMessage\(ref Message\)](#) , [Control.PreProcessControlMessage\(ref Message\)](#) ,  
[Control.ProcessKeyEventArgs\(ref Message\)](#) , [Control.ProcessKeyMessage\(ref Message\)](#) ,  
[Control.ProcessKeyPreview\(ref Message\)](#) , [Control.RaiseDragEvent\(object, DragEventArgs\)](#) ,  
[Control.RaisePaintEvent\(object, PaintEventArgs\)](#) , [Control.RecreateHandle\(\)](#) ,  
[Control.RectangleToClient\(Rectangle\)](#) , [Control.RectangleToScreen\(Rectangle\)](#) ,  
[Control.ReflectMessage\(IntPtr, ref Message\)](#) , [Control.Refresh\(\)](#) , [Control.ResetMouseEventArgs\(\)](#) ,

[Control.ResetText\(\)](#) , [Control.ResumeLayout\(\)](#) , [Control.ResumeLayout\(bool\)](#) , [Control.Scale\(SizeF\)](#) ,  
[Control.Select\(\)](#) , [Control.SelectNextControl\(Control, bool, bool, bool, bool\)](#) , [Control.SendToBack\(\)](#) ,  
[Control.SetBounds\(int, int, int, int\)](#) , [Control.SetBounds\(int, int, int, int, BoundsSpecified\)](#) ,  
[Control.SetBoundsCore\(int, int, int, int, BoundsSpecified\)](#) , [Control.SetClientSizeCore\(int, int\)](#) ,  
[Control.SizeFromClientSize\(Size\)](#) , [Control.SetStyle\(ControlStyles, bool\)](#) , [Control.SetTopLevel\(bool\)](#) ,  
[Control.SetVisibleCore\(bool\)](#) , [Control.RtlTranslateAlignment\(HorizontalAlignment\)](#) ,  
[Control.RtlTranslateAlignment\(LeftRightAlignment\)](#) ,  
[Control.RtlTranslateAlignment\(ContentAlignment\)](#) ,  
[Control.RtlTranslateHorizontal\(HorizontalAlignment\)](#) ,  
[Control.RtlTranslateLeftRight\(LeftRightAlignment\)](#) , [Control.RtlTranslateContent\(ContentAlignment\)](#) ,  
[Control.Show\(\)](#) , [Control.SuspendLayout\(\)](#) , [Control.Update\(\)](#) , [Control.UpdateBounds\(\)](#) ,  
[Control.UpdateBounds\(int, int, int, int\)](#) , [Control.UpdateBounds\(int, int, int, int, int, int\)](#) ,  
[Control.UpdateZOrder\(\)](#) , [Control.UpdateStyles\(\)](#) , [Control.OnImeModeChanged\(EventArgs\)](#) ,  
[Control.AccessibilityObject](#) , [Control.AccessibleDefaultActionDescription](#) ,  
[Control.AccessibleDescription](#) , [Control.AccessibleName](#) , [Control.AccessibleRole](#) ,  
[Control.AllowDrop](#) , [Control.Anchor](#) , [Control.AutoScrollOffset](#) , [Control.LayoutEngine](#) ,  
[Control.BackColor](#) , [Control.BackgroundImage](#) , [Control.BackgroundImageLayout](#) ,  
[Control.Bottom](#) , [Control.Bounds](#) , [Control.CanFocus](#) , [Control.CanRaiseEvents](#) ,  
[Control.CanSelect](#) , [Control.Capture](#) , [Control.CausesValidation](#) ,  
[Control.CheckForIllegalCrossThreadCalls](#) , [Control.ClientRectangle](#) , [Control.ClientSize](#) ,  
[Control.CompanyName](#) , [Control.ContainsFocus](#) , [Control.ContextMenu](#) ,  
[Control.ContextMenuStrip](#) , [Control.Controls](#) , [Control.Created](#) , [Control.Cursor](#) ,  
[Control.DataBindings](#) , [Control.DefaultBackColor](#) , [Control.DefaultCursor](#) , [Control.DefaultFont](#) ,  
[Control.DefaultForeColor](#) , [Control.DefaultMargin](#) , [Control.DefaultMaximumSize](#) ,  
[Control.DefaultMinimumSize](#) , [Control.DefaultPadding](#) , [Control.DeviceDpi](#) , [Control.IsDisposed](#) ,  
[Control.Disposing](#) , [Control.Dock](#) , [Control.DoubleBuffered](#) , [Control.Enabled](#) , [Control.Focused](#) ,  
[Control.Font](#) , [Control.FontHeight](#) , [Control.ForeColor](#) , [Control.Handle](#) , [Control.HasChildren](#) ,  
[Control.Height](#) , [Control.IsHandleCreated](#) , [Control.InvokeRequired](#) , [Control.IsAccessible](#) ,  
[Control.IsMirrored](#) , [Control.Left](#) , [Control.Location](#) , [Control.Margin](#) , [Control.MaximumSize](#) ,  
[Control.MinimumSize](#) , [Control.ModifierKeys](#) , [Control.MouseButtons](#) , [Control.mousePosition](#) ,  
[Control.Name](#) , [Control.Parent](#) , [Control.ProductName](#) , [Control.ProductVersion](#) ,  
[Control.RecreatingHandle](#) , [Control.Region](#) , [Control.ResizeRedraw](#) , [Control.Right](#) ,  
[Control.RightToLeft](#) , [Control.ScaleChildren](#) , [Control.Site](#) , [Control.Size](#) , [Control.TabIndex](#) ,  
[Control.TabStop](#) , [Control.Tag](#) , [Control.Text](#) , [Control.Top](#) , [Control.TopLevelControl](#) ,  
[Control>ShowKeyboardCues](#) , [Control>ShowFocusCues](#) , [Control.UseWaitCursor](#) , [Control.Visible](#) ,  
[Control.Width](#) , [Control.PreferredSize](#) , [Control.Padding](#) , [Control.DefaultImeMode](#) ,  
[Control.ImeMode](#) , [Control.ImeModeBase](#) , [Control.PropagatingImeMode](#) ,  
[Control.BackColorChanged](#) , [Control.BackgroundImageChanged](#) ,  
[Control.BackgroundImageLayoutChanged](#) , [Control.BindingContextChanged](#) ,  
[Control.CausesValidationChanged](#) , [Control.ClientSizeChanged](#) , [Control.ContextMenuChanged](#) ,

[Control.ContextMenuStripChanged](#) , [Control.CursorChanged](#) , [Control.DockChanged](#) ,  
[Control.EnabledChanged](#) , [Control.FontChanged](#) , [Control.ForeColorChanged](#) ,  
[Control.LocationChanged](#) , [Control.MarginChanged](#) , [Control.RegionChanged](#) ,  
[Control.RightToLeftChanged](#) , [Control.SizeChanged](#) , [Control.TabIndexChanged](#) ,  
[Control.TabStopChanged](#) , [Control.TextChanged](#) , [Control.VisibleChanged](#) , [Control.Click](#) ,  
[Control.ControlAdded](#) , [Control.ControlRemoved](#) , [Control.DragDrop](#) , [Control.DragEnter](#) ,  
[Control.DragOver](#) , [Control.DragLeave](#) , [Control.GiveFeedback](#) , [Control.HandleCreated](#) ,  
[Control.HandleDestroyed](#) , [Control.HelpRequested](#) , [Control.Invalidate](#) ,  
[Control.PaddingChanged](#) , [Control.Paint](#) , [Control.QueryContinueDrag](#) ,  
[Control.QueryAccessibilityHelp](#) , [Control.DoubleClick](#) , [Control.Enter](#) , [Control.GotFocus](#) ,  
[Control.KeyDown](#) , [Control.KeyPress](#) , [Control.KeyUp](#) , [Control.Layout](#) , [Control.Leave](#) ,  
[Control.LostFocus](#) , [Control.MouseClick](#) , [Control.MouseDoubleClick](#) ,  
[Control.MouseCaptureChanged](#) , [Control.MouseDown](#) , [Control.MouseEnter](#) ,  
[Control.MouseLeave](#) , [Control.DpiChangedBeforeParent](#) , [Control.DpiChangedAfterParent](#) ,  
[Control.MouseHover](#) , [Control.MouseMove](#) , [Control.MouseUp](#) , [Control.MouseWheel](#) ,  
[Control.Move](#) , [Control.PreviewKeyDown](#) , [Control.Resize](#) , [Control.ChangeUICues](#) ,  
[Control.StyleChanged](#) , [Control.SystemColorsChanged](#) , [Control.Validating](#) , [Control.Validated](#) ,  
[Control.ParentChanged](#) , [Control.ImeModeChanged](#) , [Component.Dispose\(\)](#) ,  
[Component.GetService\(Type\)](#) , [Component.ToString\(\)](#) , [Component.Events](#) ,  
[Component.Container](#) , [Component.DesignMode](#) , [Component.Dispose](#) ,  
[MarshalByRefObject.MemberwiseClone\(bool\)](#) , [MarshalByRefObject.GetLifetimeService\(\)](#) ,  
[MarshalByRefObject.InitializeLifetimeService\(\)](#) , [MarshalByRefObject.CreateObjRef\(Type\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### PushButtonControl(PushButton)

```
public PushButtonControl(PushButton source)
```

#### Parameters

source [PushButton](#)

## Properties

## ButtonLabel

```
public string ButtonLabel { get; set; }
```

### Property Value

[string](#) ↗

### Source

```
public PushButton Source { get; }
```

### Property Value

[PushButton](#)

## Methods

### Dispose(bool)

Clean up any resources being used.

```
protected override void Dispose(bool disposing)
```

### Parameters

**disposing** [bool](#) ↗

true if managed resources should be disposed; otherwise, false.

# Class PushButtonVisualizer

Namespace: [AllenNeuralDynamics.Core.Design](#)

Assembly: AllenNeuralDynamics.Core.Design.dll

```
public class PushButtonVisualizer : DialogTypeVisualizer
```

## Inheritance

[object](#) ← [DialogTypeVisualizer](#) ← PushButtonVisualizer

## Inherited Members

[DialogTypeVisualizer.Visualize\(IObservable<IObservable<object>>, IServiceProvider\)](#) ,  
[DialogTypeVisualizer.SequenceCompleted\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### Load(IServiceProvider)

Loads type visualizer resources using the specified service provider.

```
public override void Load(IServiceProvider provider)
```

#### Parameters

**provider** [IServiceProvider](#)

A service provider object which can be used to obtain visualization, runtime inspection, or other editing services.

### Show(object)

Updates the type visualizer to display the specified value object.

```
public override void Show(object value)
```

## Parameters

**value** [object](#)

The value to visualize.

## Unload()

Unloads all type visualizer resources.

```
public override void Unload()
```

# Class SoftwareEventVisualizer

Namespace: [AllenNeuralDynamics.Core.Design](#)

Assembly: AllenNeuralDynamics.Core.Design.dll

```
public class SoftwareEventVisualizer : BufferedVisualizer
```

## Inheritance

[object](#) ← [DialogTypeVisualizer](#) ← [BufferedVisualizer](#) ← SoftwareEventVisualizer

## Inherited Members

[BufferedVisualizer.Visualize\(IObservable<IObservable<object>>, IServiceProvider\)](#) ,  
[BufferedVisualizer.Show\(DateTime, object\)](#) , [BufferedVisualizer.TargetInterval](#) ,  
[DialogTypeVisualizer.SequenceCompleted\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### Load(IServiceProvider)

Loads type visualizer resources using the specified service provider.

```
public override void Load(IServiceProvider provider)
```

#### Parameters

provider [IServiceProvider](#)

A service provider object which can be used to obtain visualization, runtime inspection, or other editing services.

### ProcessInput(string)

```
public static string[] ProcessInput(string input)
```

Parameters

**input** [string](#)

Returns

[string](#)[]

## Show(object)

Updates the type visualizer to display the specified value object.

```
public override void Show(object value)
```

Parameters

**value** [object](#)

The value to visualize.

## ShowBuffer(IList<Timestamped<object>>)

Updates the type visualizer with a new buffer of timestamped values.

```
protected override void ShowBuffer(IList<Timestamped<object>> values)
```

Parameters

**values** [IList](#)<[Timestamped](#)<[object](#)>>

A buffer of timestamped values where each timestamp indicates the time at which the value was received.

## Unload()

Unloads all type visualizer resources.

```
public override void Unload()
```

# Class ToggleButtonStateControl

Namespace: [AllenNeuralDynamics.Core.Design](#)

Assembly: AllenNeuralDynamics.Core.Design.dll

```
public class ToggleButtonStateControl : UserControl, IDropTarget, ISynchronizeInvoke,  
IWin32Window, IBindableComponent, IComponent, IDisposable,.IContainerControl
```

## Inheritance

[object](#) ← [MarshalByRefObject](#) ← [Component](#) ← [Control](#) ← [ScrollableControl](#) ←  
[ContainerControl](#) ← [UserControl](#) ← ToggleButtonStateControl

## Implements

[IDropTarget](#), [ISynchronizeInvoke](#), [IWin32Window](#), [IBindableComponent](#), [IComponent](#),  
[IDisposable](#), [.IContainerControl](#)

## Inherited Members

[UserControl.ValidateChildren\(\)](#), [UserControl.ValidateChildren\(ValidationConstraints\)](#),  
[UserControl.OnCreateControl\(\)](#), [UserControl.OnLoad\(EventArgs\)](#),  
[UserControl.OnResize\(EventArgs\)](#), [UserControl.OnMouseDown\(MouseEventArgs\)](#),  
[UserControl.WndProc\(ref Message\)](#), [UserControl.AutoSize](#), [UserControl.AutoSizeMode](#),  
[UserControl.AutoValidate](#), [UserControl.BorderStyle](#), [UserControl.CreateParams](#),  
[UserControl.DefaultSize](#), [UserControl.AutoSizeChanged](#), [UserControl.AutoValidateChanged](#),  
[UserControl.Load](#), [ContainerControl.AdjustFormScrollbars\(bool\)](#),  
[ContainerControl.OnAutoValidateChanged\(EventArgs\)](#),  
[ContainerControl.OnFontChanged\(EventArgs\)](#), [ContainerControl.OnLayout\(LayoutEventArgs\)](#),  
[ContainerControl.OnParentChanged\(EventArgs\)](#), [ContainerControl.PerformLayout\(\)](#),  
[ContainerControl.ProcessDialogChar\(char\)](#), [ContainerControl.ProcessDialogKey\(Keys\)](#),  
[ContainerControl.ProcessCmdKey\(ref Message, Keys\)](#), [ContainerControl.ProcessMnemonic\(char\)](#),  
[ContainerControl.ProcessTabKey\(bool\)](#), [ContainerControl.Select\(bool, bool\)](#),  
[ContainerControl.UpdateDefaultButton\(\)](#), [ContainerControl.Validate\(\)](#),  
[ContainerControl.Validate\(bool\)](#), [ContainerControl.AutoScaleDimensions](#),  
[ContainerControl.AutoScaleFactor](#), [ContainerControl.AutoScaleMode](#),  
[ContainerControl.BindingContext](#), [ContainerControl.CanEnableIme](#),  
[ContainerControl.ActiveControl](#), [ContainerControl.CurrentAutoSizeDimensions](#),  
[ContainerControl.ParentForm](#), [ScrollableControl.ScrollStateAutoScrolling](#),  
[ScrollableControl.ScrollStateHScrollVisible](#), [ScrollableControl.ScrollStateVScrollVisible](#),  
[ScrollableControl.ScrollStateUserHasScrolled](#), [ScrollableControl.ScrollStateFullDrag](#),  
[ScrollableControl.GetScrollState\(int\)](#), [ScrollableControl.OnMouseWheel\(MouseEventArgs\)](#),

[ScrollableControl.OnRightToLeftChanged\(EventArgs\)](#) ,  
[ScrollableControl.OnPaintBackground\(PaintEventArgs\)](#) ,  
[ScrollableControl.OnPaddingChanged\(EventArgs\)](#) , [ScrollableControl.OnVisibleChanged\(EventArgs\)](#) ,  
[ScrollableControl.ScaleControl\(SizeF, BoundsSpecified\)](#) ,  
[ScrollableControl.SetDisplayRectLocation\(int, int\)](#) , [ScrollableControl.ScrollControlIntoView\(Control\)](#) ,  
[ScrollableControl.ScrollToControl\(Control\)](#) , [ScrollableControl.OnScroll\(ScrollEventArgs\)](#) ,  
[ScrollableControl.SetAutoScrollMargin\(int, int\)](#) , [ScrollableControl.SetScrollState\(int, bool\)](#) ,  
[ScrollableControl.AutoScroll](#) , [ScrollableControl.AutoScrollMargin](#) ,  
[ScrollableControl.AutoScrollPosition](#) , [ScrollableControl.AutoScrollMinSize](#) ,  
[ScrollableControl.DisplayRectangle](#) , [ScrollableControl.HScroll](#) , [ScrollableControl.HorizontalScroll](#) ,  
[ScrollableControl.VScroll](#) , [ScrollableControl.VerticalScroll](#) , [ScrollableControl.Scroll](#) ,  
[Control.GetAccessibilityObjectById\(int\)](#) , [Control.SetAutoSizeMode\(AutoSizeMode\)](#) ,  
[Control.GetAutoSizeMode\(\)](#) , [Control.GetPreferredSize\(Size\)](#) ,  
[Control.AccessibilityNotifyClients\(AccessibleEvents, int\)](#) ,  
[Control.AccessibilityNotifyClients\(AccessibleEvents, int, int\)](#) , [Control.BeginInvoke\(Delegate\)](#) ,  
[Control.BeginInvoke\(Delegate, params object\[\]\)](#) , [Control.BringToFront\(\)](#) ,  
[Control.Contains\(Control\)](#) , [Control.CreateAccessibilityInstance\(\)](#) , [Control.CreateControlsInstance\(\)](#) ,  
[Control.CreateGraphics\(\)](#) , [Control.CreateHandle\(\)](#) , [Control.CreateControl\(\)](#) ,  
[Control.DefWndProc\(ref Message\)](#) , [Control.DestroyHandle\(\)](#) ,  
[Control.DoDragDrop\(object, DragDropEffects\)](#) , [Control.DrawToBitmap\(Bitmap, Rectangle\)](#) ,  
[Control.EndInvoke\(IAsyncResult\)](#) , [Control.FindForm\(\)](#) , [Control.GetTopLevel\(\)](#) ,  
[Control.RaiseKeyEvent\(object, KeyEventArgs\)](#) , [Control.RaiseMouseEvent\(object, MouseEventArgs\)](#) ,  
[Control.Focus\(\)](#) , [Control.FromChildHandle\(IntPtr\)](#) , [Control.FromHandle\(IntPtr\)](#) ,  
[Control.GetChildAtPoint\(Point, GetChildAtPointSkip\)](#) , [Control.GetChildAtPoint\(Point\)](#) ,  
[Control.GetContainerControl\(\)](#) , [Control.GetScaledBounds\(Rectangle, SizeF, BoundsSpecified\)](#) ,  
[Control.GetNextControl\(Control, bool\)](#) , [Control.GetStyle\(ControlStyles\)](#) , [Control.Hide\(\)](#) ,  
[Control.InitLayout\(\)](#) , [Control.Invalidate\(Region\)](#) , [Control.Invalidate\(Region, bool\)](#) ,  
[Control.Invalidate\(\)](#) , [Control.Invalidate\(bool\)](#) , [Control.Invalidate\(Rectangle\)](#) ,  
[Control.Invalidate\(Rectangle, bool\)](#) , [Control.Invoke\(Delegate\)](#) ,  
[Control.Invoke\(Delegate, params object\[\]\)](#) , [Control.InvokePaint\(Control, PaintEventArgs\)](#) ,  
[Control.InvokePaintBackground\(Control, PaintEventArgs\)](#) , [Control.IsKeyLocked\(Keys\)](#) ,  
[Control.IsInputChar\(char\)](#) , [Control.IsInputKey\(Keys\)](#) , [Control.IsMnemonic\(char, string\)](#) ,  
[Control.LogicalToDeviceUnits\(int\)](#) , [Control.ScaleBitmapLogicalToDevice\(ref Bitmap\)](#) ,  
[Control.NotifyInvalidate\(Rectangle\)](#) , [Control.InvokeOnClick\(Control, EventArgs\)](#) ,  
[Control.OnAutoSizeChanged\(EventArgs\)](#) , [Control.OnBackColorChanged\(EventArgs\)](#) ,  
[Control.OnBackgroundImageChanged\(EventArgs\)](#) ,  
[Control.OnBackgroundImageLayoutChanged\(EventArgs\)](#) ,  
[Control.OnBindingContextChanged\(EventArgs\)](#) , [Control.OnCausesValidationChanged\(EventArgs\)](#) ,  
[Control.OnContextMenuChanged\(EventArgs\)](#) , [Control.OnContextMenuStripChanged\(EventArgs\)](#) ,  
[Control.OnCursorChanged\(EventArgs\)](#) , [Control.OnDockChanged\(EventArgs\)](#) ,

[Control.OnEnabledChanged\(EventArgs\)](#) , [Control.OnForeColorChanged\(EventArgs\)](#) ,  
[Control.OnNotifyMessage\(Message\)](#) , [Control.OnParentBackColorChanged\(EventArgs\)](#) ,  
[Control.OnParentBackgroundImageChanged\(EventArgs\)](#) ,  
[Control.OnParentBindingContextChanged\(EventArgs\)](#) , [Control.OnParentCursorChanged\(EventArgs\)](#) ,  
[Control.OnParentEnabledChanged\(EventArgs\)](#) , [Control.OnParentFontChanged\(EventArgs\)](#) ,  
[Control.OnParentForeColorChanged\(EventArgs\)](#) , [Control.OnParentRightToLeftChanged\(EventArgs\)](#) ,  
[Control.OnParentVisibleChanged\(EventArgs\)](#) , [Control.OnPrint\(PaintEventArgs\)](#) ,  
[Control.OnTabIndexChanged\(EventArgs\)](#) , [Control.OnTabStopChanged\(EventArgs\)](#) ,  
[Control.OnTextChanged\(EventArgs\)](#) , [Control.OnClick\(EventArgs\)](#) ,  
[Control.OnClientSizeChanged\(EventArgs\)](#) , [Control.OnControlAdded\(ControlEventArgs\)](#) ,  
[Control.OnControlRemoved\(ControlEventArgs\)](#) , [Control.OnHandleCreated\(EventArgs\)](#) ,  
[Control.OnLocationChanged\(EventArgs\)](#) , [Control.OnHandleDestroyed\(EventArgs\)](#) ,  
[Control.OnDoubleClick\(EventArgs\)](#) , [Control.OnDragEnter\(DragEventArgs\)](#) ,  
[Control.OnDragOver\(DragEventArgs\)](#) , [Control.OnDragLeave\(EventArgs\)](#) ,  
[Control.OnDragDrop\(DragEventArgs\)](#) , [Control.OnGiveFeedback\(GiveFeedbackEventArgs\)](#) ,  
[Control.OnEnter\(EventArgs\)](#) , [Control.InvokeGotFocus\(Control, EventArgs\)](#) ,  
[Control.OnGotFocus\(EventArgs\)](#) , [Control.OnHelpRequested\(HelpEventArgs\)](#) ,  
[Control.OnInvalidated\(InvalidEventArgs\)](#) , [Control.OnKeyDown\(KeyEventEventArgs\)](#) ,  
[Control.OnKeyPress\(KeyEventEventArgs\)](#) , [Control.OnKeyUp\(KeyEventEventArgs\)](#) ,  
[Control.OnLeave\(EventArgs\)](#) , [Control.InvokeLostFocus\(Control, EventArgs\)](#) ,  
[Control.OnLostFocus\(EventArgs\)](#) , [Control.OnMarginChanged\(EventArgs\)](#) ,  
[Control.OnMouseDoubleClick\(MouseEventArgs\)](#) , [Control.OnMouseClick\(MouseEventArgs\)](#) ,  
[Control.OnMouseCaptureChanged\(EventArgs\)](#) , [Control.OnMouseEnter\(EventArgs\)](#) ,  
[Control.OnMouseLeave\(EventArgs\)](#) , [Control.OnDpiChangedBeforeParent\(EventArgs\)](#) ,  
[Control.OnDpiChangedAfterParent\(EventArgs\)](#) , [Control.OnMouseHover\(EventArgs\)](#) ,  
[Control.OnMouseMove\(MouseEventArgs\)](#) , [Control.OnMouseUp\(MouseEventArgs\)](#) ,  
[Control.OnMove\(EventArgs\)](#) , [Control.OnPaint\(PaintEventArgs\)](#) ,  
[Control.OnQueryContinueDrag\(QueryContinueDragEventArgs\)](#) ,  
[Control.OnRegionChanged\(EventArgs\)](#) , [Control.OnPreviewKeyDown\(PreviewKeyDownEventArgs\)](#) ,  
[Control.OnSizeChanged\(EventArgs\)](#) , [Control.OnChangeUICues\(UICuesEventArgs\)](#) ,  
[Control.OnStyleChanged\(EventArgs\)](#) , [Control.OnSystemColorsChanged\(EventArgs\)](#) ,  
[Control.OnValidating\(CancelEventArgs\)](#) , [Control.OnValidated\(EventArgs\)](#) ,  
[Control.RescaleConstantsForDpi\(int, int\)](#) , [Control.PerformLayout\(\)](#) ,  
[Control.PerformLayout\(Control, string\)](#) , [Control.PointToClient\(Point\)](#) , [Control.PointToScreen\(Point\)](#) ,  
[Control.PreProcessMessage\(ref Message\)](#) , [Control.PreProcessControlMessage\(ref Message\)](#) ,  
[Control.ProcessKeyEventArgs\(ref Message\)](#) , [Control.ProcessKeyMessage\(ref Message\)](#) ,  
[Control.ProcessKeyPreview\(ref Message\)](#) , [Control.RaiseDragEvent\(object, DragEventArgs\)](#) ,  
[Control.RaisePaintEvent\(object, PaintEventArgs\)](#) , [Control.RecreateHandle\(\)](#) ,  
[Control.RectangleToClient\(Rectangle\)](#) , [Control.RectangleToScreen\(Rectangle\)](#) ,  
[Control.ReflectMessage\(IntPtr, ref Message\)](#) , [Control.Refresh\(\)](#) , [Control.ResetMouseEventArgs\(\)](#) ,

[Control.ResetText\(\)](#) , [Control.ResumeLayout\(\)](#) , [Control.ResumeLayout\(bool\)](#) , [Control.Scale\(SizeF\)](#) ,  
[Control.Select\(\)](#) , [Control.SelectNextControl\(Control, bool, bool, bool, bool\)](#) , [Control.SendToBack\(\)](#) ,  
[Control.SetBounds\(int, int, int, int\)](#) , [Control.SetBounds\(int, int, int, int, BoundsSpecified\)](#) ,  
[Control.SetBoundsCore\(int, int, int, int, BoundsSpecified\)](#) , [Control.SetClientSizeCore\(int, int\)](#) ,  
[Control.SizeFromClientSize\(Size\)](#) , [Control.SetStyle\(ControlStyles, bool\)](#) , [Control.SetTopLevel\(bool\)](#) ,  
[Control.SetVisibleCore\(bool\)](#) , [Control.RtlTranslateAlignment\(HorizontalAlignment\)](#) ,  
[Control.RtlTranslateAlignment\(LeftRightAlignment\)](#) ,  
[Control.RtlTranslateAlignment\(ContentAlignment\)](#) ,  
[Control.RtlTranslateHorizontal\(HorizontalAlignment\)](#) ,  
[Control.RtlTranslateLeftRight\(LeftRightAlignment\)](#) , [Control.RtlTranslateContent\(ContentAlignment\)](#) ,  
[Control.Show\(\)](#) , [Control.SuspendLayout\(\)](#) , [Control.Update\(\)](#) , [Control.UpdateBounds\(\)](#) ,  
[Control.UpdateBounds\(int, int, int, int\)](#) , [Control.UpdateBounds\(int, int, int, int, int, int\)](#) ,  
[Control.UpdateZOrder\(\)](#) , [Control.UpdateStyles\(\)](#) , [Control.OnImeModeChanged\(EventArgs\)](#) ,  
[Control.AccessibilityObject](#) , [Control.AccessibleDefaultActionDescription](#) ,  
[Control.AccessibleDescription](#) , [Control.AccessibleName](#) , [Control.AccessibleRole](#) ,  
[Control.AllowDrop](#) , [Control.Anchor](#) , [Control.AutoScrollOffset](#) , [Control.LayoutEngine](#) ,  
[Control.BackColor](#) , [Control.BackgroundImage](#) , [Control.BackgroundImageLayout](#) ,  
[Control.Bottom](#) , [Control.Bounds](#) , [Control.CanFocus](#) , [Control.CanRaiseEvents](#) ,  
[Control.CanSelect](#) , [Control.Capture](#) , [Control.CausesValidation](#) ,  
[Control.CheckForIllegalCrossThreadCalls](#) , [Control.ClientRectangle](#) , [Control.ClientSize](#) ,  
[Control.CompanyName](#) , [Control.ContainsFocus](#) , [Control.ContextMenu](#) ,  
[Control.ContextMenuStrip](#) , [Control.Controls](#) , [Control.Created](#) , [Control.Cursor](#) ,  
[Control.DataBindings](#) , [Control.DefaultBackColor](#) , [Control.DefaultCursor](#) , [Control.DefaultFont](#) ,  
[Control.DefaultForeColor](#) , [Control.DefaultMargin](#) , [Control.DefaultMaximumSize](#) ,  
[Control.DefaultMinimumSize](#) , [Control.DefaultPadding](#) , [Control.DeviceDpi](#) , [Control.IsDisposed](#) ,  
[Control.Disposing](#) , [Control.Dock](#) , [Control.DoubleBuffered](#) , [Control.Enabled](#) , [Control.Focused](#) ,  
[Control.Font](#) , [Control.FontHeight](#) , [Control.ForeColor](#) , [Control.Handle](#) , [Control.HasChildren](#) ,  
[Control.Height](#) , [Control.IsHandleCreated](#) , [Control.InvokeRequired](#) , [Control.IsAccessible](#) ,  
[Control.IsMirrored](#) , [Control.Left](#) , [Control.Location](#) , [Control.Margin](#) , [Control.MaximumSize](#) ,  
[Control.MinimumSize](#) , [Control.ModifierKeys](#) , [Control.MouseButtons](#) , [Control.mousePosition](#) ,  
[Control.Name](#) , [Control.Parent](#) , [Control.ProductName](#) , [Control.ProductVersion](#) ,  
[Control.RecreatingHandle](#) , [Control.Region](#) , [Control.ResizeRedraw](#) , [Control.Right](#) ,  
[Control.RightToLeft](#) , [Control.ScaleChildren](#) , [Control.Site](#) , [Control.Size](#) , [Control.TabIndex](#) ,  
[Control.TabStop](#) , [Control.Tag](#) , [Control.Text](#) , [Control.Top](#) , [Control.TopLevelControl](#) ,  
[Control>ShowKeyboardCues](#) , [Control>ShowFocusCues](#) , [Control.UseWaitCursor](#) , [Control.Visible](#) ,  
[Control.Width](#) , [Control.PreferredSize](#) , [Control.Padding](#) , [Control.DefaultImeMode](#) ,  
[Control.ImeMode](#) , [Control.ImeModeBase](#) , [Control.PropagatingImeMode](#) ,  
[Control.BackColorChanged](#) , [Control.BackgroundImageChanged](#) ,  
[Control.BackgroundImageLayoutChanged](#) , [Control.BindingContextChanged](#) ,  
[Control.CausesValidationChanged](#) , [Control.ClientSizeChanged](#) , [Control.ContextMenuChanged](#) ,

[Control.ContextMenuStripChanged](#) , [Control.CursorChanged](#) , [Control.DockChanged](#) ,  
[Control.EnabledChanged](#) , [Control.FontChanged](#) , [Control.ForeColorChanged](#) ,  
[Control.LocationChanged](#) , [Control.MarginChanged](#) , [Control.RegionChanged](#) ,  
[Control.RightToLeftChanged](#) , [Control.SizeChanged](#) , [Control.TabIndexChanged](#) ,  
[Control.TabStopChanged](#) , [Control.TextChanged](#) , [Control.VisibleChanged](#) , [Control.Click](#) ,  
[Control.ControlAdded](#) , [Control.ControlRemoved](#) , [Control.DragDrop](#) , [Control.DragEnter](#) ,  
[Control.DragOver](#) , [Control.DragLeave](#) , [Control.GiveFeedback](#) , [Control.HandleCreated](#) ,  
[Control.HandleDestroyed](#) , [Control.HelpRequested](#) , [Control.Invalidate](#) ,  
[Control.PaddingChanged](#) , [Control.Paint](#) , [Control.QueryContinueDrag](#) ,  
[Control.QueryAccessibilityHelp](#) , [Control.DoubleClick](#) , [Control.Enter](#) , [Control.GotFocus](#) ,  
[Control.KeyDown](#) , [Control.KeyPress](#) , [Control.KeyUp](#) , [Control.Layout](#) , [Control.Leave](#) ,  
[Control.LostFocus](#) , [Control.MouseClick](#) , [Control.MouseDoubleClick](#) ,  
[Control.MouseCaptureChanged](#) , [Control.MouseDown](#) , [Control.MouseEnter](#) ,  
[Control.MouseLeave](#) , [Control.DpiChangedBeforeParent](#) , [Control.DpiChangedAfterParent](#) ,  
[Control.MouseHover](#) , [Control.MouseMove](#) , [Control.MouseUp](#) , [Control.MouseWheel](#) ,  
[Control.Move](#) , [Control.PreviewKeyDown](#) , [Control.Resize](#) , [Control.ChangeUICues](#) ,  
[Control.StyleChanged](#) , [Control.SystemColorsChanged](#) , [Control.Validating](#) , [Control.Validated](#) ,  
[Control.ParentChanged](#) , [Control.ImeModeChanged](#) , [Component.Dispose\(\)](#) ,  
[Component.GetService\(Type\)](#) , [Component.ToString\(\)](#) , [Component.Events](#) ,  
[Component.Container](#) , [Component.DesignMode](#) , [Component.Dispose](#) ,  
[MarshalByRefObject.MemberwiseClone\(bool\)](#) , [MarshalByRefObject.GetLifetimeService\(\)](#) ,  
[MarshalByRefObject.InitializeLifetimeService\(\)](#) , [MarshalByRefObject.CreateObjRef\(Type\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### ToggleButtonStateControl(ToggleButton)

```
public ToggleButtonStateControl(ToggleButton source)
```

#### Parameters

source [ToggleButton](#)

## Properties

## CheckedLabel

```
public string CheckedLabel { get; set; }
```

Property Value

[string ↗](#)

## Source

```
public ToggleButton Source { get; }
```

Property Value

[ToggleButton](#)

## State

```
public bool State { get; set; }
```

Property Value

[bool ↗](#)

## UncheckedLabel

```
public string UncheckedLabel { get; set; }
```

Property Value

[string ↗](#)

## Methods

## Dispose(bool)

Clean up any resources being used.

```
protected override void Dispose(bool disposing)
```

Parameters

**disposing** [bool](#)

true if managed resources should be disposed; otherwise, false.

# Class ToggleButtonVisualizer

Namespace: [AllenNeuralDynamics.Core.Design](#)

Assembly: AllenNeuralDynamics.Core.Design.dll

```
public class ToggleButtonVisualizer : DialogTypeVisualizer
```

## Inheritance

[object](#) ← [DialogTypeVisualizer](#) ← ToggleButtonVisualizer

## Inherited Members

[DialogTypeVisualizer.Visualize\(IObservable<IObservable<object>>, IServiceProvider\)](#) ,  
[DialogTypeVisualizer.SequenceCompleted\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### Load(IServiceProvider)

Loads type visualizer resources using the specified service provider.

```
public override void Load(IServiceProvider provider)
```

#### Parameters

**provider** [IServiceProvider](#)

A service provider object which can be used to obtain visualization, runtime inspection, or other editing services.

### Show(object)

Updates the type visualizer to display the specified value object.

```
public override void Show(object value)
```

## Parameters

**value** [object](#)

The value to visualize.

## Unload()

Unloads all type visualizer resources.

```
public override void Unload()
```

# Class ToggleEnabledEventArgs

Namespace: [AllenNeuralDynamics.Core.Design](#)

Assembly: AllenNeuralDynamics.Core.Design.dll

```
public class ToggleEnabledEventArgs : EventArgs
```

## Inheritance

[object](#) ← [EventArgs](#) ← ToggleEnabledEventArgs

## Inherited Members

[EventArgs.Empty](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Enabled

```
public bool Enabled { get; set; }
```

## Property Value

[bool](#)

# Namespace AllenNeuralDynamics.Cuttlefish

## Classes

### [ArmExternalStartTrigger](#)

Represents a register that if set to 1, the device will execute the PMW task using the selected pins.

### [ArmExternalStopTrigger](#)

Represents a register that if set to 1, the device will stop the PMW task using the selected pins.

### [AsyncDevice](#)

Represents an asynchronous API to configure and interface with Cuttlefish devices.

### [ConfigurePwm](#)

Represents an operator that generates a sequence of Harp messages to configure the PWM feature.

### [CreateArmExternalStartTriggerPayload](#)

Represents an operator that creates a message payload that if set to 1, the device will execute the PMW task using the selected pins.

### [CreateArmExternalStopTriggerPayload](#)

Represents an operator that creates a message payload that if set to 1, the device will stop the PMW task using the selected pins.

### [CreateExternalStartTriggerEdgePayload](#)

Represents an operator that creates a message payload that set the edge of the external trigger. 0: Rising, 1: Falling.

### [CreateExternalStopTriggerEdgePayload](#)

Represents an operator that creates a message payload that set the edge of the external trigger. 0: Rising, 1: Falling.

### [CreateMessage](#)

Represents an operator which creates standard message payloads for the Cuttlefish device.

### [CreatePortDirectionPayload](#)

Represents an operator that creates a message payload that set the direction of the ports.

### [CreatePortStatePayload](#)

Represents an operator that creates a message payload that read or write the state of the ports. An event will be triggered when the state changes without a write command.

### [CreatePwmTaskPayload](#)

Represents an operator that creates a message payload that struct to configure the PWM task.  
offset\_us (U32), start\_time\_us (U32), stop\_time\_us (U32), port\_mask (U8), cycles (U32), invert (U8).

#### [CreateSoftwareStartTriggerPayload](#)

Represents an operator that creates a message payload that writing a non-0 value to this register will trigger the PWM task.

#### [CreateSoftwareStopTriggerPayload](#)

Represents an operator that creates a message payload that writing a non-0 value to this register will stop the PWM task.

#### [CreateTaskControlPayload](#)

Represents an operator that creates a message payload for register TaskControl.

#### [CreateTimestampedArmExternalStartTriggerPayload](#)

Represents an operator that creates a timestamped message payload that if set to 1, the device will execute the PMW task using the selected pins.

#### [CreateTimestampedArmExternalStopTriggerPayload](#)

Represents an operator that creates a timestamped message payload that if set to 1, the device will stop the PMW task using the selected pins.

#### [CreateTimestampedExternalStartTriggerEdgePayload](#)

Represents an operator that creates a timestamped message payload that set the edge of the external trigger. 0: Rising, 1: Falling.

#### [CreateTimestampedExternalStopTriggerEdgePayload](#)

Represents an operator that creates a timestamped message payload that set the edge of the external trigger. 0: Rising, 1: Falling.

#### [CreateTimestampedPortDirectionPayload](#)

Represents an operator that creates a timestamped message payload that set the direction of the ports.

#### [CreateTimestampedPortStatePayload](#)

Represents an operator that creates a timestamped message payload that read or write the state of the ports. An event will be triggered when the state changes without a write command.

#### [CreateTimestampedPwmTaskPayload](#)

Represents an operator that creates a timestamped message payload that struct to configure the PWM task. offset\_us (U32), start\_time\_us (U32), stop\_time\_us (U32), port\_mask (U8), cycles (U32), invert (U8).

#### [CreateTimestampedSoftwareStartTriggerPayload](#)

Represents an operator that creates a timestamped message payload that writing a non-0 value to this register will trigger the PWM task.

### [CreateTimestampedSoftwareStopTriggerPayload](#)

Represents an operator that creates a timestamped message payload that writing a non-0 value to this register will stop the PWM task.

### [CreateTimestampedTaskControlPayload](#)

Represents an operator that creates a timestamped message payload for register TaskControl.

### [Device](#)

Represents an observable source of messages from the Harp device connected at the specified serial port.

### [ExternalStartTriggerEdge](#)

Represents a register that set the edge of the external trigger. 0: Rising, 1: Falling.

### [ExternalStopTriggerEdge](#)

Represents a register that set the edge of the external trigger. 0: Rising, 1: Falling.

### [FilterRegister](#)

Represents an operator that filters register-specific messages reported by the [AllenNeuralDynamics.Cuttlefish](#) device.

### [Format](#)

Represents an operator which formats a sequence of values as specific Cuttlefish register messages.

### [GetMetadata](#)

Represents an operator that returns the contents of the metadata file describing the [AllenNeuralDynamics.Cuttlefish](#) device registers.

### [GroupByRegister](#)

Represents an operator that groups the sequence of [AllenNeuralDynamics.Cuttlefish](#) messages by register type.

### [HelperMethods](#)

### [Parse](#)

Represents an operator which filters and selects specific messages reported by the Cuttlefish device.

### [PortDirection](#)

Represents a register that set the direction of the ports.

### [PortState](#)

Represents a register that read or write the state of the ports. An event will be triggered when the state changes without a write command.

### [PwmTask](#)

Represents a register that struct to configure the PWM task. offset\_us (U32), start\_time\_us (U32), stop\_time\_us (U32), port\_mask (U8), cycles (U32), invert (U8).

### [SoftwareStartTrigger](#)

Represents a register that writing a non-0 value to this register will trigger the PWM task.

### [SoftwareStopTrigger](#)

Represents a register that writing a non-0 value to this register will stop the PWM task.

### [TaskControl](#)

Represents a register that manipulates messages from register TaskControl.

### [TimestampedArmExternalStartTrigger](#)

Provides methods for manipulating timestamped messages from the ArmExternalStartTrigger register.

### [TimestampedArmExternalStopTrigger](#)

Provides methods for manipulating timestamped messages from the ArmExternalStopTrigger register.

### [TimestampedExternalStartTriggerEdge](#)

Provides methods for manipulating timestamped messages from the ExternalStartTriggerEdge register.

### [TimestampedExternalStopTriggerEdge](#)

Provides methods for manipulating timestamped messages from the ExternalStopTriggerEdge register.

### [TimestampedPortDirection](#)

Provides methods for manipulating timestamped messages from the PortDirection register.

### [TimestampedPortState](#)

Provides methods for manipulating timestamped messages from the PortState register.

### [TimestampedPwmTask](#)

Provides methods for manipulating timestamped messages from the PwmTask register.

### [TimestampedSoftwareStartTrigger](#)

Provides methods for manipulating timestamped messages from the SoftwareStartTrigger register.

### [TimestampedSoftwareStopTrigger](#)

Provides methods for manipulating timestamped messages from the SoftwareStopTrigger register.

### [TimestampedTaskControl](#)

Provides methods for manipulating timestamped messages from the TaskControl register.

## Structs

### [TaskControlPayload](#)

Represents the payload of the TaskControl register.

## Enums

### [Ports](#)

Available ports on the device

# Class ArmExternalStartTrigger

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents a register that if set to 1, the device will execute the PMW task using the selected pins.

```
public class ArmExternalStartTrigger
```

## Inheritance

[object](#) ← ArmExternalStartTrigger

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [ArmExternalStartTrigger](#) register. This field is constant.

```
public const int Address = 35
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [ArmExternalStartTrigger](#) register. This field is constant.

```
public const int RegisterLength = 1
```

### Field Value

## .RegisterType

Represents the payload type of the [ArmExternalStartTrigger](#) register. This field is constant.

```
public const PayloadType RegisterType = U8
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, Ports)

Returns a Harp message for the [ArmExternalStartTrigger](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, Ports value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [Ports](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [ArmExternalStartTrigger](#) register with the specified message type and payload.

## FromPayload(double, MessageType, Ports)

Returns a timestamped Harp message for the [ArmExternalStartTrigger](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
Ports value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [Ports](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [ArmExternalStartTrigger](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [ArmExternalStartTrigger](#) register messages.

```
public static Ports GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## Ports

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [ArmExternalStartTrigger](#) register messages.

```
public static Timestamped<Ports> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <Ports>

A value representing the timestamped message payload.

# Class ArmExternalStopTrigger

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents a register that if set to 1, the device will stop the PMW task using the selected pins.

```
public class ArmExternalStopTrigger
```

## Inheritance

[object](#) ← ArmExternalStopTrigger

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [ArmExternalStopTrigger](#) register. This field is constant.

```
public const int Address = 37
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [ArmExternalStopTrigger](#) register. This field is constant.

```
public const int RegisterLength = 1
```

### Field Value

## .RegisterType

Represents the payload type of the [ArmExternalStopTrigger](#) register. This field is constant.

```
public const PayloadType RegisterType = U8
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, Ports)

Returns a Harp message for the [ArmExternalStopTrigger](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, Ports value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [Ports](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [ArmExternalStopTrigger](#) register with the specified message type and payload.

## FromPayload(double, MessageType, Ports)

Returns a timestamped Harp message for the [ArmExternalStopTrigger](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
Ports value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [Ports](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [ArmExternalStopTrigger](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [ArmExternalStopTrigger](#) register messages.

```
public static Ports GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## Ports

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [ArmExternalStopTrigger](#) register messages.

```
public static Timestamped<Ports> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <Ports>

A value representing the timestamped message payload.

# Class AsyncDevice

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an asynchronous API to configure and interface with Cuttlefish devices.

```
public class AsyncDevice : AsyncDevice, IDisposable
```

## Inheritance

[object](#) ← [AsyncDevice](#) ← AsyncDevice

## Implements

[IDisposable](#)

## Inherited Members

[AsyncDevice.ReadWhoAmIAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadHardwareVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadAssemblyVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadCoreVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadFirmwareVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadTimestampSecondsAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadTimestampMicrosecondsAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadOperationControlAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadResetDeviceAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadDeviceNameAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadSerialNumberAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadClockConfigurationAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadByteAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadByteArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSByteAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSByteArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt16Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt16ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt16Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt16ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt32Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt32ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt32Async\(int, CancellationToken\)](#) ,

[AsyncDevice.ReadInt32ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt64Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt64ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt64Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt64ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSingleAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSingleArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.WriteTimestampSecondsAsync\(uint, CancellationToken\)](#) ,  
[AsyncDevice.WriteResetDeviceAsync\(ResetFlags, CancellationToken\)](#) ,  
[AsyncDevice.WriteDeviceNameAsync\(string, CancellationToken\)](#) ,  
[AsyncDevice.WriteClockConfigurationAsync\(ClockConfigurationFlags, CancellationToken\)](#) ,  
[AsyncDevice.WriteByteAsync\(int, byte, CancellationToken\)](#) ,  
[AsyncDevice.WriteByteAsync\(int, byte\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteSByteAsync\(int, sbyte, CancellationToken\)](#) ,  
[AsyncDevice.WriteSByteAsync\(int, sbyte\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteUInt16Async\(int, ushort, CancellationToken\)](#) ,  
[AsyncDevice.WriteUInt16Async\(int, ushort\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine16Async\(int, short, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine16Async\(int, short\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, uint, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, uint\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, int, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, int\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, ulong, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, ulong\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, long, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, long\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteSingleAsync\(int, float, CancellationToken\)](#) ,  
[AsyncDevice.WriteSingleAsync\(int, float\[\], CancellationToken\)](#) ,  
[AsyncDevice.CommandAsync\(HarpMessage, CancellationToken\)](#) , [AsyncDevice.Dispose\(\)](#) ,  
[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Methods

### ReadArmExternalStartTriggerAsync(CancellationToken)

Asynchronously reads the contents of the ArmExternalStartTrigger register.

```
public Task<Ports> ReadArmExternalStartTriggerAsync(CancellationToken cancellationToken = default)
```

## Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<Ports>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadArmExternalStopTriggerAsync(CancellationToken)

Asynchronously reads the contents of the ArmExternalStopTrigger register.

```
public Task<Ports> ReadArmExternalStopTriggerAsync(CancellationToken cancellationToken = default)
```

## Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<Ports>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadExternalStartTriggerEdgeAsync(CancellationToken)

Asynchronously reads the contents of the ExternalStartTriggerEdge register.

```
public Task<Ports> ReadExternalStartTriggerEdgeAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<Ports>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadExternalStopTriggerEdgeAsync(CancellationToken)

Asynchronously reads the contents of the ExternalStopTriggerEdge register.

```
public Task<Ports> ReadExternalStopTriggerEdgeAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<Ports>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadPortDirectionAsync(CancellationToken)

Asynchronously reads the contents of the PortDirection register.

```
public Task<Ports> ReadPortDirectionAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<Ports>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadPortStateAsync(CancellationToken)

Asynchronously reads the contents of the PortState register.

```
public Task<Ports> ReadPortStateAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<Ports>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadPwmTaskAsync(CancellationToken)

Asynchronously reads the contents of the PwmTask register.

```
public Task<byte[]> ReadPwmTaskAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[byte](#)[]>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadSoftwareStartTriggerAsync(CancellationToken)

Asynchronously reads the contents of the SoftwareStartTrigger register.

```
public Task<byte> ReadSoftwareStartTriggerAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[byte](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadSoftwareStopTriggerAsync(CancellationToken)

Asynchronously reads the contents of the SoftwareStopTrigger register.

```
public Task<byte> ReadSoftwareStopTriggerAsync(CancellationToken cancellationToken  
= default)
```

## Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#) <[byte](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadTaskControlAsync(CancellationToken)

Asynchronously reads the contents of the TaskControl register.

```
public Task<TaskControlPayload> ReadTaskControlAsync(CancellationToken cancellationToken  
= default)
```

## Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#) <[TaskControlPayload](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadTimestampedArmExternalStartTriggerAsync(CancellationToken)

Asynchronously reads the timestamped contents of the ArmExternalStartTrigger register.

```
public Task<Timestamped<Ports>>
ReadTimestampedArmExternalStartTriggerAsync(CancellationToken cancellationToken = default)
```

## Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<[Ports](#)>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedArmExternalStopTriggerAsync(CancellationTokenToken)

Asynchronously reads the timestamped contents of the ArmExternalStopTrigger register.

```
public Task<Timestamped<Ports>> ReadTimestampedArmExternalStopTriggerAsync(CancellationToken
cancellationToken = default)
```

## Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<[Ports](#)>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedExternalStartTriggerEdgeAsync(CancellationToken)

Asynchronously reads the timestamped contents of the ExternalStartTriggerEdge register.

```
public Task<Timestamped<Ports>>
ReadTimestampedExternalStartTriggerEdgeAsync(CancellationToken cancellationToken = default)
```

### Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

### Returns

[Task](#)<[Timestamped](#)<Ports>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedExternalStopTriggerEdgeAsync(CancellationToken)

Asynchronously reads the timestamped contents of the ExternalStopTriggerEdge register.

```
public Task<Timestamped<Ports>>
ReadTimestampedExternalStopTriggerEdgeAsync(CancellationToken cancellationToken = default)
```

### Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

### Returns

[Task](#)<[Timestamped](#)<Ports>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedPortDirectionAsync(CancellationToken)

Asynchronously reads the timestamped contents of the PortDirection register.

```
public Task<Timestamped<Ports>> ReadTimestampedPortDirectionAsync(CancellationToken  
cancellationToken = default)
```

### Parameters

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

### Returns

[Task](#)<[Timestamped](#)<Ports>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedPortStateAsync(CancellationToken)

Asynchronously reads the timestamped contents of the PortState register.

```
public Task<Timestamped<Ports>> ReadTimestampedPortStateAsync(CancellationToken  
cancellationToken = default)
```

### Parameters

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

### Returns

[Task](#)<[Timestamped](#)<Ports>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedPwmTaskAsync(CancellationToken)

Asynchronously reads the timestamped contents of the PwmTask register.

```
public Task<Timestamped<byte[]>> ReadTimestampedPwmTaskAsync(CancellationToken  
cancellationToken = default)
```

### Parameters

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

### Returns

[Task](#)<[Timestamped](#)<[byte](#)>[]>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedSoftwareStartTriggerAsync(CancellationToken)

Asynchronously reads the timestamped contents of the SoftwareStartTrigger register.

```
public Task<Timestamped<byte>> ReadTimestampedSoftwareStartTriggerAsync(CancellationToken  
cancellationToken = default)
```

### Parameters

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

### Returns

## [Task](#)<[Timestamped](#)<[byte](#)>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedSoftwareStopTriggerAsync(CancellationToken)

Asynchronously reads the timestamped contents of the SoftwareStopTrigger register.

```
public Task<Timestamped<byte>> ReadTimestampedSoftwareStopTriggerAsync(CancellationToken cancellationToken = default)
```

### Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

### Returns

## [Task](#)<[Timestamped](#)<[byte](#)>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedTaskControlAsync(CancellationToken)

Asynchronously reads the timestamped contents of the TaskControl register.

```
public Task<Timestamped<TaskControlPayload>>
ReadTimestampedTaskControlAsync(CancellationToken cancellationToken = default)
```

### Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#) <[Timestamped](#) <[TaskControlPayload](#)>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## WriteArmExternalStartTriggerAsync(Ports, CancellationToken)

Asynchronously writes a value to the ArmExternalStartTrigger register.

```
public Task WriteArmExternalStartTriggerAsync(Ports value, CancellationToken  
cancellationToken = default)
```

## Parameters

**value** [Ports](#)

The value to be stored in the register.

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)

The task object representing the asynchronous write operation.

## WriteArmExternalStopTriggerAsync(Ports, CancellationToken)

Asynchronously writes a value to the ArmExternalStopTrigger register.

```
public Task WriteArmExternalStopTriggerAsync(Ports value, CancellationToken  
cancellationToken = default)
```

## Parameters

**value** [Ports](#)

The value to be stored in the register.

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

Returns

[Task](#)

The task object representing the asynchronous write operation.

## WriteExternalStartTriggerEdgeAsync(Ports, CancellationToken)

Asynchronously writes a value to the ExternalStartTriggerEdge register.

```
public Task WriteExternalStartTriggerEdgeAsync(Ports value, CancellationToken  
cancellationToken = default)
```

Parameters

`value` [Ports](#)

The value to be stored in the register.

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

Returns

[Task](#)

The task object representing the asynchronous write operation.

## WriteExternalStopTriggerEdgeAsync(Ports, CancellationToken)

Asynchronously writes a value to the ExternalStopTriggerEdge register.

```
public Task WriteExternalStopTriggerEdgeAsync(Ports value, CancellationToken
```

```
cancellationToken = default)
```

## Parameters

### `value` [Ports](#)

The value to be stored in the register.

### `cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

### [Task](#)

The task object representing the asynchronous write operation.

## WritePortDirectionAsync(Ports, CancellationToken)

Asynchronously writes a value to the PortDirection register.

```
public Task WritePortDirectionAsync(Ports value, CancellationToken cancellationToken  
= default)
```

## Parameters

### `value` [Ports](#)

The value to be stored in the register.

### `cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

### [Task](#)

The task object representing the asynchronous write operation.

## WritePortStateAsync(Ports, CancellationToken)

Asynchronously writes a value to the PortState register.

```
public Task WritePortStateAsync(Ports value, CancellationToken cancellationToken = default)
```

Parameters

**value** [Ports](#)

The value to be stored in the register.

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

Returns

[Task](#)

The task object representing the asynchronous write operation.

## WritePwmTaskAsync(byte[], CancellationToken)

Asynchronously writes a value to the PwmTask register.

```
public Task WritePwmTaskAsync(byte[] value, CancellationToken cancellationToken = default)
```

Parameters

**value** [byte](#)[]

The value to be stored in the register.

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

Returns

[Task](#)

The task object representing the asynchronous write operation.

## WriteSoftwareStartTriggerAsync(byte, CancellationToken)

Asynchronously writes a value to the SoftwareStartTrigger register.

```
public Task WriteSoftwareStartTriggerAsync(byte value, CancellationToken cancellationToken  
= default)
```

### Parameters

**value** [byte](#)

The value to be stored in the register.

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

### Returns

[Task](#)

The task object representing the asynchronous write operation.

## WriteSoftwareStopTriggerAsync(byte, CancellationToken)

Asynchronously writes a value to the SoftwareStopTrigger register.

```
public Task WriteSoftwareStopTriggerAsync(byte value, CancellationToken cancellationToken  
= default)
```

### Parameters

**value** [byte](#)

The value to be stored in the register.

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

Returns

[Task](#)

The task object representing the asynchronous write operation.

## WriteTaskControlAsync(TaskControlPayload, CancellationToken)

Asynchronously writes a value to the TaskControl register.

```
public Task WriteTaskControlAsync(TaskControlPayload value, CancellationToken  
cancellationToken = default)
```

Parameters

**value** [TaskControlPayload](#)

The value to be stored in the register.

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

Returns

[Task](#)

The task object representing the asynchronous write operation.

# Class ConfigurePwm

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that generates a sequence of Harp messages to configure the PWM feature.

```
public class ConfigurePwm : Source<HarpMessage>
```

## Inheritance

[object](#) ← [Source](#)<[HarpMessage](#)> ← ConfigurePwm

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Address

Gets or sets the address of the Harp Message

```
public int Address { get; set; }
```

## Property Value

[int](#)

## Delay

Gets or sets the PWM protocol delay.

```
public uint Delay { get; set; }
```

## Property Value

[uint](#)

## Invert

Gets or sets a value specifying whether generation of the PWM should be inverted.

```
public bool Invert { get; set; }
```

Property Value

[bool](#)

## MessageType

Gets or sets the type of the Harp Message

```
public MessageType MessageType { get; set; }
```

Property Value

[MessageType](#)

## OnTime

Gets or sets the on-time of the PWM pulse. Defined in microseconds.

```
public uint OnTime { get; set; }
```

Property Value

[uint](#)

## Period

Gets or sets the period of the PWM pulse. Defined in microseconds.

```
public uint Period { get; set; }
```

Property Value

[uint](#)

## Port

Gets or sets the number of pulses to trigger on the specified PWM. If the default value of zero is specified, the PWM will be infinite.

```
public Ports Port { get; set; }
```

Property Value

[Ports](#)

## RepeatCount

Gets or sets the number of times the PWM protocol will be repeated.

```
public uint RepeatCount { get; set; }
```

Property Value

[uint](#)

## Methods

### BuildMessage(int, MessageType, double?)

Builds a message to configure the PWM task.

```
public HarpMessage BuildMessage(int address, MessageType messageType, double? timestamp = null)
```

## Parameters

address [int](#)

messageType [MessageType](#)

timestamp [double](#)?

## Returns

[HarpMessage](#)

## Generate()

Generates an observable sequence of Harp messages to configure a PWM task.

```
public override IObservable<HarpMessage> Generate()
```

## Returns

[IObservable](#)<[HarpMessage](#)>

A sequence of [HarpMessage](#) objects representing a command to configure a PWM task.

## Generate<TSource>(IObservable<TSource>)

Generates an observable sequence of Harp messages to configure the PWM feature whenever the source sequence emits a notification.

```
public IObservable<HarpMessage> Generate<TSource>(IObservable<TSource> source)
```

## Parameters

source [IObservable](#)<TSource>

The sequence containing the notifications used to emit new configuration messages.

## Returns

## [IObservable](#) <[HarpMessage](#)>

A sequence of [HarpMessage](#) objects representing the commands needed to fully configure the PWM feature.

### Type Parameters

#### TSource

The type of the elements in the `source` sequence.

# Class CreateArmExternalStartTriggerPayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that creates a message payload that if set to 1, the device will execute the PMW task using the selected pins.

```
public class CreateArmExternalStartTriggerPayload
```

## Inheritance

[object](#) ← CreateArmExternalStartTriggerPayload

## Derived

[CreateTimestampedArmExternalStartTriggerPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## ArmExternalStartTrigger

Gets or sets the value that if set to 1, the device will execute the PMW task using the selected pins.

```
public Ports ArmExternalStartTrigger { get; set; }
```

## Property Value

[Ports](#)

# Methods

## GetMessage(MessageType)

Creates a message that if set to 1, the device will execute the PMW task using the selected pins.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the ArmExternalStartTrigger register.

## GetPayload()

Creates a message payload for the ArmExternalStartTrigger register.

```
public Ports GetPayload()
```

## Returns

[Ports](#)

The created message payload value.

# Class CreateArmExternalStopTriggerPayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that creates a message payload that if set to 1, the device will stop the PMW task using the selected pins.

```
public class CreateArmExternalStopTriggerPayload
```

## Inheritance

[object](#) ← CreateArmExternalStopTriggerPayload

## Derived

[CreateTimestampedArmExternalStopTriggerPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## ArmExternalStopTrigger

Gets or sets the value that if set to 1, the device will stop the PMW task using the selected pins.

```
public Ports ArmExternalStopTrigger { get; set; }
```

## Property Value

[Ports](#)

# Methods

## GetMessage(MessageType)

Creates a message that if set to 1, the device will stop the PMW task using the selected pins.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the ArmExternalStopTrigger register.

## GetPayload()

Creates a message payload for the ArmExternalStopTrigger register.

```
public Ports GetPayload()
```

## Returns

[Ports](#)

The created message payload value.

# Class CreateExternalStartTriggerEdgePayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that creates a message payload that set the edge of the external trigger. 0: Rising, 1: Falling.

```
public class CreateExternalStartTriggerEdgePayload
```

## Inheritance

[object](#) ← CreateExternalStartTriggerEdgePayload

## Derived

[CreateTimestampedExternalStartTriggerEdgePayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## ExternalStartTriggerEdge

Gets or sets the value that set the edge of the external trigger. 0: Rising, 1: Falling.

```
public Ports ExternalStartTriggerEdge { get; set; }
```

## Property Value

[Ports](#)

# Methods

## GetMessage(MessageType)

Creates a message that set the edge of the external trigger. 0: Rising, 1: Falling.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the ExternalStartTriggerEdge register.

## GetPayload()

Creates a message payload for the ExternalStartTriggerEdge register.

```
public Ports GetPayload()
```

## Returns

[Ports](#)

The created message payload value.

# Class CreateExternalStopTriggerEdgePayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that creates a message payload that set the edge of the external trigger. 0: Rising, 1: Falling.

```
public class CreateExternalStopTriggerEdgePayload
```

## Inheritance

[object](#) ← CreateExternalStopTriggerEdgePayload

## Derived

[CreateTimestampedExternalStopTriggerEdgePayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Properties

### ExternalStopTriggerEdge

Gets or sets the value that set the edge of the external trigger. 0: Rising, 1: Falling.

```
public Ports ExternalStopTriggerEdge { get; set; }
```

## Property Value

[Ports](#)

## Methods

### GetMessage(MessageType)

Creates a message that set the edge of the external trigger. 0: Rising, 1: Falling.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the ExternalStopTriggerEdge register.

## GetPayload()

Creates a message payload for the ExternalStopTriggerEdge register.

```
public Ports GetPayload()
```

## Returns

[Ports](#)

The created message payload value.

# Class CreateMessage

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator which creates standard message payloads for the Cuttlefish device.

```
public class CreateMessage : CreateMessageBuilder, IExpressionBuilder,  
ICustomTypeDescriptor, INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [CreateMessageBuilder](#) ← CreateMessage

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[CreateMessageBuilder.Build\(IEnumerable<Expression>\)](#), [CreateMessageBuilder.ArgumentRange](#),  
[CreateMessageBuilder.MessageType](#), [CreateMessageBuilder.Payload](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

## Constructors

### CreateMessage()

Initializes a new instance of the [CreateMessage](#) class.

```
public CreateMessage()
```

## See Also

[CreatePortDirectionPayload](#)

[CreatePortStatePayload](#)

[CreatePwmTaskPayload](#)

[CreateArmExternalStartTriggerPayload](#)

[CreateExternalStartTriggerEdgePayload](#)

[CreateArmExternalStopTriggerPayload](#)

[CreateExternalStopTriggerEdgePayload](#)

[CreateSoftwareStartTriggerPayload](#)

[CreateSoftwareStopTriggerPayload](#)

[CreateTaskControlPayload](#)

# Class CreatePortDirectionPayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that creates a message payload that set the direction of the ports.

```
public class CreatePortDirectionPayload
```

## Inheritance

[object](#) ← CreatePortDirectionPayload

## Derived

[CreateTimestampedPortDirectionPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Properties

### PortDirection

Gets or sets the value that set the direction of the ports.

```
public Ports PortDirection { get; set; }
```

## Property Value

[Ports](#)

## Methods

### GetMessage(MessageType)

Creates a message that set the direction of the ports.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the PortDirection register.

## GetPayload()

Creates a message payload for the PortDirection register.

```
public Ports GetPayload()
```

## Returns

[Ports](#)

The created message payload value.

# Class CreatePortStatePayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that creates a message payload that read or write the state of the ports. An event will be triggered when the state changes without a write command.

```
public class CreatePortStatePayload
```

## Inheritance

[object](#) ← CreatePortStatePayload

## Derived

[CreateTimestampedPortStatePayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## PortState

Gets or sets the value that read or write the state of the ports. An event will be triggered when the state changes without a write command.

```
public Ports PortState { get; set; }
```

## Property Value

[Ports](#)

# Methods

## GetMessage(MessageType)

Creates a message that read or write the state of the ports. An event will be triggered when the state changes without a write command.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the PortState register.

## GetPayload()

Creates a message payload for the PortState register.

```
public Ports GetPayload()
```

## Returns

[Ports](#)

The created message payload value.

# Class CreatePwmTaskPayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that creates a message payload that struct to configure the PWM task. offset\_us (U32), start\_time\_us (U32), stop\_time\_us (U32), port\_mask (U8), cycles (U32), invert (U8).

```
public class CreatePwmTaskPayload
```

## Inheritance

[object](#) ← CreatePwmTaskPayload

## Derived

[CreateTimestampedPwmTaskPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## PwmTask

Gets or sets the value that struct to configure the PWM task. offset\_us (U32), start\_time\_us (U32), stop\_time\_us (U32), port\_mask (U8), cycles (U32), invert (U8).

```
public byte[] PwmTask { get; set; }
```

## Property Value

[byte](#)[]

# Methods

## GetMessage(MessageType)

Creates a message that struct to configure the PWM task. offset\_us (U32), start\_time\_us (U32), stop\_time\_us (U32), port\_mask (U8), cycles (U32), invert (U8).

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the PwmTask register.

## GetPayload()

Creates a message payload for the PwmTask register.

```
public byte[] GetPayload()
```

## Returns

[byte](#)[]

The created message payload value.

# Class CreateSoftwareStartTriggerPayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that creates a message payload that writing a non-0 value to this register will trigger the PWM task.

```
public class CreateSoftwareStartTriggerPayload
```

## Inheritance

[object](#) ← CreateSoftwareStartTriggerPayload

## Derived

[CreateTimestampedSoftwareStartTriggerPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## SoftwareStartTrigger

Gets or sets the value that writing a non-0 value to this register will trigger the PWM task.

```
public byte SoftwareStartTrigger { get; set; }
```

## Property Value

[byte](#)

# Methods

## GetMessage(MessageType)

Creates a message that writing a non-0 value to this register will trigger the PWM task.

```
public HarpMessage GetMessage(MessageType messageType)
```

Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

Returns

[HarpMessage](#)

A new message for the SoftwareStartTrigger register.

## GetPayload()

Creates a message payload for the SoftwareStartTrigger register.

```
public byte GetPayload()
```

Returns

[byte](#)

The created message payload value.

# Class CreateSoftwareStopTriggerPayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that creates a message payload that writing a non-0 value to this register will stop the PWM task.

```
public class CreateSoftwareStopTriggerPayload
```

## Inheritance

[object](#) ← CreateSoftwareStopTriggerPayload

## Derived

[CreateTimestampedSoftwareStopTriggerPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## SoftwareStopTrigger

Gets or sets the value that writing a non-0 value to this register will stop the PWM task.

```
public byte SoftwareStopTrigger { get; set; }
```

## Property Value

[byte](#)

# Methods

## GetMessage(MessageType)

Creates a message that writing a non-0 value to this register will stop the PWM task.

```
public HarpMessage GetMessage(MessageType messageType)
```

Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

Returns

[HarpMessage](#)

A new message for the SoftwareStopTrigger register.

## GetPayload()

Creates a message payload for the SoftwareStopTrigger register.

```
public byte GetPayload()
```

Returns

[byte](#)

The created message payload value.

# Class CreateTaskControlPayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that creates a message payload for register TaskControl.

```
public class CreateTaskControlPayload
```

## Inheritance

[object](#) ← CreateTaskControlPayload

## Derived

[CreateTimestampedTaskControlPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## ClearAllTasks

Gets or sets a value that halts and clears all tasks.

```
public EnableFlag ClearAllTasks { get; set; }
```

## Property Value

[EnableFlag](#)

## DumpAllTasks

Gets or sets a value that sends an event from PwmTask register per currently configured task. Once all events have been sent, a write message will be returned from this register.

```
public EnableFlag DumpAllTasks { get; set; }
```

## Property Value

[EnableFlag](#) ↗

## TaskCount

Gets or sets a value that number of tasks currently configured. This portion of the register is read-only.

```
public byte TaskCount { get; set; }
```

## Property Value

[byte](#) ↗

## Methods

### GetMessage(MessageType)

Creates a message for register TaskControl.

```
public HarpMessage GetMessage(MessageType messageType)
```

#### Parameters

**messageType** [MessageType](#) ↗

Specifies the type of the created message.

#### Returns

[HarpMessage](#) ↗

A new message for the TaskControl register.

## GetPayload()

Creates a message payload for the TaskControl register.

```
public TaskControlPayload GetPayload()
```

Returns

[TaskControlPayload](#)

The created message payload value.

# Class

# CreateTimestampedArmExternalStartTriggerPayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that creates a timestamped message payload that if set to 1, the device will execute the PMW task using the selected pins.

```
public class CreateTimestampedArmExternalStartTriggerPayload :  
CreateArmExternalStartTriggerPayload
```

## Inheritance

[object](#) ← [CreateArmExternalStartTriggerPayload](#) ← CreateTimestampedArmExternalStartTriggerPayload

## Inherited Members

[CreateArmExternalStartTriggerPayload.ArmExternalStartTrigger](#) ,  
[CreateArmExternalStartTriggerPayload.GetPayload\(\)](#) ,  
[CreateArmExternalStartTriggerPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that if set to 1, the device will execute the PMW task using the selected pins.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

Returns

[HarpMessage](#)

A new timestamped message for the ArmExternalStartTrigger register.

# Class

# CreateTimestampedArmExternalStopTriggerPayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that creates a timestamped message payload that if set to 1, the device will stop the PMW task using the selected pins.

```
public class CreateTimestampedArmExternalStopTriggerPayload :  
CreateArmExternalStopTriggerPayload
```

## Inheritance

[object](#) ← [CreateArmExternalStopTriggerPayload](#) ← CreateTimestampedArmExternalStopTriggerPayload

## Inherited Members

[CreateArmExternalStopTriggerPayload.ArmExternalStopTrigger](#) ,  
[CreateArmExternalStopTriggerPayload.GetPayload\(\)](#) ,  
[CreateArmExternalStopTriggerPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that if set to 1, the device will stop the PMW task using the selected pins.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

Returns

[HarpMessage](#)

A new timestamped message for the ArmExternalStopTrigger register.

# Class

# CreateTimestampedExternalStartTriggerEdgePayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that creates a timestamped message payload that set the edge of the external trigger. 0: Rising, 1: Falling.

```
public class CreateTimestampedExternalStartTriggerEdgePayload :  
CreateExternalStartTriggerEdgePayload
```

## Inheritance

[object](#) ← [CreateExternalStartTriggerEdgePayload](#) ←  
CreateTimestampedExternalStartTriggerEdgePayload

## Inherited Members

[CreateExternalStartTriggerEdgePayload.ExternalStartTriggerEdge](#) ,  
[CreateExternalStartTriggerEdgePayload.GetPayload\(\)](#) ,  
[CreateExternalStartTriggerEdgePayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that set the edge of the external trigger. 0: Rising, 1: Falling.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

Returns

[HarpMessage](#)

A new timestamped message for the ExternalStartTriggerEdge register.

# Class

# CreateTimestampedExternalStopTriggerEdgePayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that creates a timestamped message payload that set the edge of the external trigger. 0: Rising, 1: Falling.

```
public class CreateTimestampedExternalStopTriggerEdgePayload :  
CreateExternalStopTriggerEdgePayload
```

## Inheritance

[object](#) ← [CreateExternalStopTriggerEdgePayload](#) ←  
CreateTimestampedExternalStopTriggerEdgePayload

## Inherited Members

[CreateExternalStopTriggerEdgePayload.ExternalStopTriggerEdge](#) ,  
[CreateExternalStopTriggerEdgePayload.GetPayload\(\)](#) ,  
[CreateExternalStopTriggerEdgePayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that set the edge of the external trigger. 0: Rising, 1: Falling.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

Returns

[HarpMessage](#)

A new timestamped message for the ExternalStopTriggerEdge register.

# Class CreateTimestampedPortDirectionPayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that creates a timestamped message payload that set the direction of the ports.

```
public class CreateTimestampedPortDirectionPayload : CreatePortDirectionPayload
```

## Inheritance

[object](#) ← [CreatePortDirectionPayload](#) ← CreateTimestampedPortDirectionPayload

## Inherited Members

[CreatePortDirectionPayload.PortDirection](#) , [CreatePortDirectionPayload.GetPayload\(\)](#) ,  
[CreatePortDirectionPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that set the direction of the ports.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

#### Returns

[HarpMessage](#)

A new timestamped message for the PortDirection register.

# Class CreateTimestampedPortStatePayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that creates a timestamped message payload that read or write the state of the ports. An event will be triggered when the state changes without a write command.

```
public class CreateTimestampedPortStatePayload : CreatePortStatePayload
```

## Inheritance

[object](#) ← [CreatePortStatePayload](#) ← CreateTimestampedPortStatePayload

## Inherited Members

[CreatePortStatePayload.PortState](#) , [CreatePortStatePayload.GetPayload\(\)](#) ,  
[CreatePortStatePayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that read or write the state of the ports. An event will be triggered when the state changes without a write command.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new timestamped message for the PortState register.

# Class CreateTimestampedPwmTaskPayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that creates a timestamped message payload that struct to configure the PWM task. offset\_us (U32), start\_time\_us (U32), stop\_time\_us (U32), port\_mask (U8), cycles (U32), invert (U8).

```
public class CreateTimestampedPwmTaskPayload : CreatePwmTaskPayload
```

## Inheritance

[object](#) ← [CreatePwmTaskPayload](#) ← CreateTimestampedPwmTaskPayload

## Inherited Members

[CreatePwmTaskPayload.PwmTask](#) , [CreatePwmTaskPayload.GetPayload\(\)](#) ,  
[CreatePwmTaskPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that struct to configure the PWM task. offset\_us (U32), start\_time\_us (U32), stop\_time\_us (U32), port\_mask (U8), cycles (U32), invert (U8).

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new timestamped message for the PwmTask register.

# Class

## CreateTimestampedSoftwareStartTriggerPayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that creates a timestamped message payload that writing a non-0 value to this register will trigger the PWM task.

```
public class CreateTimestampedSoftwareStartTriggerPayload :  
CreateSoftwareStartTriggerPayload
```

### Inheritance

[object](#) ← [CreateSoftwareStartTriggerPayload](#) ← CreateTimestampedSoftwareStartTriggerPayload

### Inherited Members

[CreateSoftwareStartTriggerPayload.SoftwareStartTrigger](#) ,  
[CreateSoftwareStartTriggerPayload.GetPayload\(\)](#) ,  
[CreateSoftwareStartTriggerPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that writing a non-0 value to this register will trigger the PWM task.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

### Parameters

timestamp [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

Returns

[HarpMessage](#)

A new timestamped message for the SoftwareStartTrigger register.

# Class

# CreateTimestampedSoftwareStopTriggerPayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that creates a timestamped message payload that writing a non-0 value to this register will stop the PWM task.

```
public class CreateTimestampedSoftwareStopTriggerPayload : CreateSoftwareStopTriggerPayload
```

## Inheritance

[object](#) ← [CreateSoftwareStopTriggerPayload](#) ← CreateTimestampedSoftwareStopTriggerPayload

## Inherited Members

[CreateSoftwareStopTriggerPayload.SoftwareStopTrigger](#) ,  
[CreateSoftwareStopTriggerPayload.GetPayload\(\)](#) ,  
[CreateSoftwareStopTriggerPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

# Methods

## GetMessage(double, MessageType)

Creates a timestamped message that writing a non-0 value to this register will stop the PWM task.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

Returns

[HarpMessage](#) ↗

A new timestamped message for the SoftwareStopTrigger register.

# Class CreateTimestampedTaskControlPayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that creates a timestamped message payload for register TaskControl.

```
public class CreateTimestampedTaskControlPayload : CreateTaskControlPayload
```

## Inheritance

[object](#) ← [CreateTaskControlPayload](#) ← CreateTimestampedTaskControlPayload

## Inherited Members

[CreateTaskControlPayload.ClearAllTasks](#) , [CreateTaskControlPayload.DumpAllTasks](#) ,  
[CreateTaskControlPayload.TaskCount](#) , [CreateTaskControlPayload.GetPayload\(\)](#) ,  
[CreateTaskControlPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message for register TaskControl.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

#### Returns

## [HarpMessage](#)

A new timestamped message for the TaskControl register.

# Class Device

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an observable source of messages from the Harp device connected at the specified serial port.

```
[Combinator(MethodName = "Generate")]
[WorkflowElementCategory(ElementCategory.Source)]
public class Device : Device, INamedElement
```

## Inheritance

[object](#) ← [Source](#) < [HarpMessage](#) > ← [Device](#) ← [Device](#)

## Implements

[INamedElement](#)

## Inherited Members

[Device.Generate\(\)](#) , [Device.Generate\(IObservable<HarpMessage>\)](#) , [Device.OperationMode](#) ,  
[Device.OperationLed](#) , [Device.DumpRegisters](#) , [Device.VisualIndicators](#) , [Device.Heartbeat](#) ,  
[Device.IgnoreErrors](#) , [Device.PortName](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

# Constructors

## Device()

Initializes a new instance of the [Device](#) class.

```
public Device()
```

# Fields

## Metadata

Gets the contents of the metadata file describing the [AllenNeuralDynamics.Cuttlefish](#) device registers.

```
public static readonly string Metadata
```

Field Value

[string](#)

## WhoAmI

Represents the unique identity class of the [AllenNeuralDynamics.Cuttlefish](#) device. This field is constant.

```
public const int WhoAmI = 1403
```

Field Value

[int](#)

## Properties

### RegisterMap

Gets a read-only mapping from address to register type.

```
public static IReadOnlyDictionary<int, Type> RegisterMap { get; }
```

Property Value

[IReadOnlyDictionary](#)<[int](#), [Type](#)>

## Methods

### CreateAsync(string)

Initializes a new instance of the asynchronous API to configure and interface with Cuttlefish devices on the specified serial port.

```
public static Task<AsyncDevice> CreateAsync(string portName)
```

## Parameters

**portName** [string](#)

The name of the serial port used to communicate with the Harp device.

## Returns

[Task](#) <[AsyncDevice](#)>

A task that represents the asynchronous initialization operation. The value of the [Result](#) parameter contains a new instance of the [AsyncDevice](#) class.

# Class ExternalStartTriggerEdge

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents a register that set the edge of the external trigger. 0: Rising, 1: Falling.

```
public class ExternalStartTriggerEdge
```

## Inheritance

[object](#) ← ExternalStartTriggerEdge

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [ExternalStartTriggerEdge](#) register. This field is constant.

```
public const int Address = 36
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [ExternalStartTriggerEdge](#) register. This field is constant.

```
public const int RegisterLength = 1
```

### Field Value

## .RegisterType

Represents the payload type of the [ExternalStartTriggerEdge](#) register. This field is constant.

```
public const PayloadType RegisterType = U8
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, Ports)

Returns a Harp message for the [ExternalStartTriggerEdge](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, Ports value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [Ports](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [ExternalStartTriggerEdge](#) register with the specified message type and payload.

## FromPayload(double, MessageType, Ports)

Returns a timestamped Harp message for the [ExternalStartTriggerEdge](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
Ports value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [Ports](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [ExternalStartTriggerEdge](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [ExternalStartTriggerEdge](#) register messages.

```
public static Ports GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## Ports

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [ExternalStartTriggerEdge](#) register messages.

```
public static Timestamped<Ports> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <Ports>

A value representing the timestamped message payload.

# Class ExternalStopTriggerEdge

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents a register that set the edge of the external trigger. 0: Rising, 1: Falling.

```
public class ExternalStopTriggerEdge
```

## Inheritance

[object](#) ← ExternalStopTriggerEdge

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [ExternalStopTriggerEdge](#) register. This field is constant.

```
public const int Address = 38
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [ExternalStopTriggerEdge](#) register. This field is constant.

```
public const int RegisterLength = 1
```

### Field Value

## .RegisterType

Represents the payload type of the [ExternalStopTriggerEdge](#) register. This field is constant.

```
public const PayloadType RegisterType = U8
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, Ports)

Returns a Harp message for the [ExternalStopTriggerEdge](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, Ports value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [Ports](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [ExternalStopTriggerEdge](#) register with the specified message type and payload.

## FromPayload(double, MessageType, Ports)

Returns a timestamped Harp message for the [ExternalStopTriggerEdge](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
Ports value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [Ports](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [ExternalStopTriggerEdge](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [ExternalStopTriggerEdge](#) register messages.

```
public static Ports GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## Ports

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [ExternalStopTriggerEdge](#) register messages.

```
public static Timestamped<Ports> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <Ports>

A value representing the timestamped message payload.

# Class FilterRegister

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that filters register-specific messages reported by the [AllenNeuralDynamics.Cuttlefish](#) device.

```
public class FilterRegister : FilterRegisterBuilder, IExpressionBuilder,  
ICustomTypeDescriptor, INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [RegisterCombinatorBuilder](#) ← [FilterRegisterBuilder](#) ← FilterRegister

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[FilterRegisterBuilder.FilterType](#), [FilterRegisterBuilder.Register](#),  
[RegisterCombinatorBuilder.Build\(IEnumerable<Expression>\)](#),  
[RegisterCombinatorBuilder.ArgumentRange](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

## Constructors

### FilterRegister()

Initializes a new instance of the [FilterRegister](#) class.

```
public FilterRegister()
```

## See Also

[PortDirection](#)

[PortState](#)

[PwmTask](#)

[ArmExternalStartTrigger](#)

[ExternalStartTriggerEdge](#)

[ArmExternalStopTrigger](#)

[ExternalStopTriggerEdge](#)

[SoftwareStartTrigger](#)

[SoftwareStopTrigger](#)

[TaskControl](#)

# Class Format

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator which formats a sequence of values as specific Cuttlefish register messages.

```
public class Format : FormatBuilder, IExpressionBuilder, ICustomTypeDescriptor,  
INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [FormatBuilder](#) ← Format

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[FormatBuilder.Build\(IEnumerable<Expression>\)](#), [FormatBuilder.ArgumentRange](#),  
[FormatBuilder.MessageType](#), [FormatBuilder.Register](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

## Constructors

### Format()

Initializes a new instance of the [Format](#) class.

```
public Format()
```

## See Also

[PortDirection](#)

[PortState](#)

[PwmTask](#)

[ArmExternalStartTrigger](#)

[ExternalStartTriggerEdge](#)

[ArmExternalStopTrigger](#)

[ExternalStopTriggerEdge](#)

[SoftwareStartTrigger](#)

[SoftwareStopTrigger](#)

[TaskControl](#)

# Class GetMetadata

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that returns the contents of the metadata file describing the [AllenNeuralDynamics.Cuttlefish](#) device registers.

```
public class GetMetadata : Source<string>
```

## Inheritance

[object](#) ← [Source](#)<[string](#)> ← GetMetadata

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Methods

### Generate()

Returns an observable sequence with the contents of the metadata file describing the [AllenNeuralDynamics.Cuttlefish](#) device registers.

```
public override IObservable<string> Generate()
```

## Returns

[IObservable](#)<[string](#)>

A sequence with a single [string](#) object representing the contents of the metadata file.

# Class GroupByRegister

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator that groups the sequence of [AllenNeuralDynamics.Cuttlefish](#)" messages by register type.

```
public class GroupByRegister : Combinator<HarpMessage, IGroupedObservable<Type, HarpMessage>>
```

## Inheritance

```
object ↗ ← Combinator ↗ <HarpMessage ↗, IGroupedObservable ↗ <Type ↗, HarpMessage ↗>> ← GroupByRegister
```

## Inherited Members

```
Combinator<HarpMessage, IGroupedObservable<Type, HarpMessage>>.Process(I Observable<HarpMessage>) ↗ ,  
object.ToString() ↗ , object.Equals(object) ↗ , object.Equals(object, object) ↗ ,  
object.ReferenceEquals(object, object) ↗ , object.GetHashCode() ↗ , object.GetType() ↗ ,  
object.MemberwiseClone() ↗
```

## Methods

### Process(I Observable<HarpMessage>)

Groups an observable sequence of [AllenNeuralDynamics.Cuttlefish](#) messages by register type.

```
public override IObservable<IGroupedObservable<Type, HarpMessage>>  
Process(IObservable<HarpMessage> source)
```

## Parameters

source [IObservable](#)<HarpMessage>

The sequence of Harp device messages.

## Returns

[IObservable](#) < [IGroupedObservable](#) < [Type](#), [HarpMessage](#) >>

A sequence of observable groups, each of which corresponds to a unique [AllenNeuralDynamics.Cuttlefish](#) register.

# Class HelperMethods

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

```
public class HelperMethods
```

## Inheritance

[object](#) ← HelperMethods

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Class Parse

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents an operator which filters and selects specific messages reported by the Cuttlefish device.

```
public class Parse : ParseBuilder, IExpressionBuilder, ICustomTypeDescriptor, INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [RegisterCombinatorBuilder](#) ← [ParseBuilder](#) ← Parse

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[ParseBuilder.Register](#), [RegisterCombinatorBuilder.Build\(IEnumerable<Expression>\)](#),  
[RegisterCombinatorBuilder.ArgumentRange](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

## Constructors

### Parse()

Initializes a new instance of the [Parse](#) class.

```
public Parse()
```

## See Also

[PortDirection](#)

[PortState](#)

[PwmTask](#)

[ArmExternalStartTrigger](#)

[ExternalStartTriggerEdge](#)

[ArmExternalStopTrigger](#)

[ExternalStopTriggerEdge](#)

[SoftwareStartTrigger](#)

[SoftwareStopTrigger](#)

[TaskControl](#)

# Class PortDirection

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents a register that set the direction of the ports.

```
public class PortDirection
```

## Inheritance

[object](#) ← PortDirection

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [PortDirection](#) register. This field is constant.

```
public const int Address = 32
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [PortDirection](#) register. This field is constant.

```
public const int RegisterLength = 1
```

### Field Value

## .RegisterType

Represents the payload type of the [PortDirection](#) register. This field is constant.

```
public const PayloadType RegisterType = U8
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, Ports)

Returns a Harp message for the [PortDirection](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, Ports value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [Ports](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [PortDirection](#) register with the specified message type and payload.

### FromPayload(double, MessageType, Ports)

Returns a timestamped Harp message for the [PortDirection](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
Ports value)
```

Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [Ports](#)

The value to be stored in the message payload.

Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [PortDirection](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [PortDirection](#) register messages.

```
public static Ports GetPayload(HarpMessage message)
```

Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

Returns

[Ports](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [PortDirection](#) register messages.

```
public static Timestamped<Ports> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <Ports>

A value representing the timestamped message payload.

# Class PortState

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents a register that read or write the state of the ports. An event will be triggered when the state changes without a write command.

```
public class PortState
```

## Inheritance

[object](#) ← PortState

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Fields

## Address

Represents the address of the [PortState](#) register. This field is constant.

```
public const int Address = 33
```

## Field Value

[int](#)

## RegisterLength

Represents the length of the [PortState](#) register. This field is constant.

```
public const int RegisterLength = 1
```

## Field Value

[int](#)

## RegisterType

Represents the payload type of the [PortState](#) register. This field is constant.

```
public const PayloadType RegisterType = U8
```

## Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, Ports)

Returns a Harp message for the [PortState](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, Ports value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [Ports](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [PortState](#) register with the specified message type and payload.

## FromPayload(double, MessageType, Ports)

Returns a timestamped Harp message for the [PortState](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
Ports value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [Ports](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [PortState](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [PortState](#) register messages.

```
public static Ports GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## [Ports](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [PortState](#) register messages.

```
public static Timestamped<Ports> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <Ports>

A value representing the timestamped message payload.

# Enum Ports

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Available ports on the device

```
[Flags]  
public enum Ports : byte
```

## Fields

None = 0

Port0 = 1

Port1 = 2

Port2 = 4

Port3 = 8

Port4 = 16

Port5 = 32

Port6 = 64

Port7 = 128

# Class PwmTask

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents a register that struct to configure the PWM task. offset\_us (U32), start\_time\_us (U32), stop\_time\_us (U32), port\_mask (U8), cycles (U32), invert (U8).

```
public class PwmTask
```

## Inheritance

[object](#) ← PwmTask

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [PwmTask](#) register. This field is constant.

```
public const int Address = 34
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [PwmTask](#) register. This field is constant.

```
public const int RegisterLength = 18
```

## Field Value

[int↗](#)

## RegisterType

Represents the payload type of the [PwmTask](#) register. This field is constant.

```
public const PayloadType RegisterType = U8
```

## Field Value

[PayloadType↗](#)

## Methods

### FromPayload(MessageType, byte[])

Returns a Harp message for the [PwmTask](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, byte[] value)
```

#### Parameters

**messageType** [MessageType↗](#)

The type of the Harp message.

**value** [byte↗\[\]](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage↗](#)

A [HarpMessage↗](#) object for the [PwmTask](#) register with the specified message type and payload.

## FromPayload(double, MessageType, byte[])

Returns a timestamped Harp message for the [PwmTask](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,
byte[] value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [byte](#)[]

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [PwmTask](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [PwmTask](#) register messages.

```
public static byte[] GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[byte](#)[]

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [PwmTask](#) register messages.

```
public static Timestamped<byte[]> GetTimestampedPayload(HarpMessage message)
```

### Parameters

[message](#) [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#)<[byte](#)[]>

A value representing the timestamped message payload.

# Class SoftwareStartTrigger

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents a register that writing a non-0 value to this register will trigger the PWM task.

```
public class SoftwareStartTrigger
```

## Inheritance

[object](#) ← SoftwareStartTrigger

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [SoftwareStartTrigger](#) register. This field is constant.

```
public const int Address = 39
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [SoftwareStartTrigger](#) register. This field is constant.

```
public const int RegisterLength = 1
```

### Field Value

## RegisterType

Represents the payload type of the [SoftwareStartTrigger](#) register. This field is constant.

```
public const PayloadType RegisterType = U8
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, byte)

Returns a Harp message for the [SoftwareStartTrigger](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, byte value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [byte](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [SoftwareStartTrigger](#) register with the specified message type and payload.

## FromPayload(double, MessageType, byte)

Returns a timestamped Harp message for the [SoftwareStartTrigger](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType, byte value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [byte](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [SoftwareStartTrigger](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [SoftwareStartTrigger](#) register messages.

```
public static byte GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## [byte](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [SoftwareStartTrigger](#) register messages.

```
public static Timestamped<byte> GetTimestampedPayload(HarpMessage message)
```

### Parameters

[message](#) [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <[byte](#)>

A value representing the timestamped message payload.

# Class SoftwareStopTrigger

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents a register that writing a non-0 value to this register will stop the PWM task.

```
public class SoftwareStopTrigger
```

## Inheritance

[object](#) ← SoftwareStopTrigger

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [SoftwareStopTrigger](#) register. This field is constant.

```
public const int Address = 40
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [SoftwareStopTrigger](#) register. This field is constant.

```
public const int RegisterLength = 1
```

### Field Value

## .RegisterType

Represents the payload type of the [SoftwareStopTrigger](#) register. This field is constant.

```
public const PayloadType RegisterType = U8
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, byte)

Returns a Harp message for the [SoftwareStopTrigger](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, byte value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [byte](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [SoftwareStopTrigger](#) register with the specified message type and payload.

## FromPayload(double, MessageType, byte)

Returns a timestamped Harp message for the [SoftwareStopTrigger](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType, byte value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [byte](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [SoftwareStopTrigger](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [SoftwareStopTrigger](#) register messages.

```
public static byte GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## [byte](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [SoftwareStopTrigger](#) register messages.

```
public static Timestamped<byte> GetTimestampedPayload(HarpMessage message)
```

### Parameters

[message](#) [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <[byte](#)>

A value representing the timestamped message payload.

# Class TaskControl

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents a register that manipulates messages from register TaskControl.

```
public class TaskControl
```

## Inheritance

[object](#) ← TaskControl

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [TaskControl](#) register. This field is constant.

```
public const int Address = 41
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [TaskControl](#) register. This field is constant.

```
public const int RegisterLength = 1
```

### Field Value

## .RegisterType

Represents the payload type of the [TaskControl](#) register. This field is constant.

```
public const PayloadType RegisterType = U8
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, TaskControlPayload)

Returns a Harp message for the [TaskControl](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, TaskControlPayload value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [TaskControlPayload](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [TaskControl](#) register with the specified message type and payload.

### FromPayload(double, MessageType, TaskControlPayload)

Returns a timestamped Harp message for the [TaskControl](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
TaskControlPayload value)
```

Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [TaskControlPayload](#)

The value to be stored in the message payload.

Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [TaskControl](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [TaskControl](#) register messages.

```
public static TaskControlPayload GetPayload(HarpMessage message)
```

Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

Returns

[TaskControlPayload](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [TaskControl](#) register messages.

```
public static Timestamped<TaskControlPayload> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <[TaskControlPayload](#)>

A value representing the timestamped message payload.

# Struct TaskControlPayload

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Represents the payload of the TaskControl register.

```
public struct TaskControlPayload
```

## Inherited Members

[ValueType.Equals\(object\)](#) , [ValueType.GetHashCode\(\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetType\(\)](#)

## Constructors

### TaskControlPayload(EnableFlag, EnableFlag, byte)

Initializes a new instance of the [TaskControlPayload](#) structure.

```
public TaskControlPayload(EnableFlag clearAllTasks, EnableFlag dumpAllTasks, byte taskCount)
```

## Parameters

**clearAllTasks** [EnableFlag](#)

Halts and clears all tasks.

**dumpAllTasks** [EnableFlag](#)

Sends an event from PwmTask register per currently configured task. Once all events have been sent, a write message will be returned from this register.

**taskCount** [byte](#)

Number of tasks currently configured. This portion of the register is read-only.

## Fields

## ClearAllTasks

Halts and clears all tasks.

```
public EnableFlag ClearAllTasks
```

Field Value

[EnableFlag](#) ↗

## DumpAllTasks

Sends an event from PwmTask register per currently configured task. Once all events have been sent, a write message will be returned from this register.

```
public EnableFlag DumpAllTasks
```

Field Value

[EnableFlag](#) ↗

## TaskCount

Number of tasks currently configured. This portion of the register is read-only.

```
public byte TaskCount
```

Field Value

[byte](#) ↗

## Methods

### ToString()

Returns a [string](#) ↗ that represents the payload of the TaskControl register.

```
public override string ToString()
```

Returns

[string](#)

A [string](#) that represents the payload of the TaskControl register.

# Class TimestampedArmExternalStartTrigger

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Provides methods for manipulating timestamped messages from the ArmExternalStartTrigger register.

```
public class TimestampedArmExternalStartTrigger
```

## Inheritance

[object](#) ← TimestampedArmExternalStartTrigger

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [ArmExternalStartTrigger](#) register. This field is constant.

```
public const int Address = 35
```

### Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [ArmExternalStartTrigger](#) register messages.

```
public static Timestamped<Ports> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <Ports>

A value representing the timestamped message payload.

## See Also

[ArmExternalStartTrigger](#)

# Class TimestampedArmExternalStopTrigger

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Provides methods for manipulating timestamped messages from the ArmExternalStopTrigger register.

```
public class TimestampedArmExternalStopTrigger
```

## Inheritance

[object](#) ← TimestampedArmExternalStopTrigger

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [ArmExternalStopTrigger](#) register. This field is constant.

```
public const int Address = 37
```

## Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [ArmExternalStopTrigger](#) register messages.

```
public static Timestamped<Ports> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <Ports>

A value representing the timestamped message payload.

## See Also

[ArmExternalStopTrigger](#)

# Class TimestampedExternalStartTriggerEdge

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Provides methods for manipulating timestamped messages from the ExternalStartTriggerEdge register.

```
public class TimestampedExternalStartTriggerEdge
```

## Inheritance

[object](#) ← TimestampedExternalStartTriggerEdge

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [ExternalStartTriggerEdge](#) register. This field is constant.

```
public const int Address = 36
```

## Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [ExternalStartTriggerEdge](#) register messages.

```
public static Timestamped<Ports> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <Ports>

A value representing the timestamped message payload.

## See Also

[ExternalStartTriggerEdge](#)

# Class TimestampedExternalStopTriggerEdge

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Provides methods for manipulating timestamped messages from the ExternalStopTriggerEdge register.

```
public class TimestampedExternalStopTriggerEdge
```

## Inheritance

[object](#) ← TimestampedExternalStopTriggerEdge

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [ExternalStopTriggerEdge](#) register. This field is constant.

```
public const int Address = 38
```

### Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [ExternalStopTriggerEdge](#) register messages.

```
public static Timestamped<Ports> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <Ports>

A value representing the timestamped message payload.

## See Also

[ExternalStopTriggerEdge](#)

# Class TimestampedPortDirection

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Provides methods for manipulating timestamped messages from the PortDirection register.

```
public class TimestampedPortDirection
```

## Inheritance

[object](#) ← TimestampedPortDirection

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [PortDirection](#) register. This field is constant.

```
public const int Address = 32
```

## Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [PortDirection](#) register messages.

```
public static Timestamped<Ports> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <Ports>

A value representing the timestamped message payload.

## See Also

[PortDirection](#)

# Class TimestampedPortState

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Provides methods for manipulating timestamped messages from the PortState register.

```
public class TimestampedPortState
```

## Inheritance

[object](#) ← TimestampedPortState

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [PortState](#) register. This field is constant.

```
public const int Address = 33
```

## Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [PortState](#) register messages.

```
public static Timestamped<Ports> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[Ports](#)>

A value representing the timestamped message payload.

## See Also

[PortState](#)

# Class TimestampedPwmTask

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Provides methods for manipulating timestamped messages from the PwmTask register.

```
public class TimestampedPwmTask
```

## Inheritance

[object](#) ← TimestampedPwmTask

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [PwmTask](#) register. This field is constant.

```
public const int Address = 34
```

## Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [PwmTask](#) register messages.

```
public static Timestamped<byte[]> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#)<[byte](#)[]>

A value representing the timestamped message payload.

## See Also

[PwmTask](#)

# Class TimestampedSoftwareStartTrigger

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Provides methods for manipulating timestamped messages from the SoftwareStartTrigger register.

```
public class TimestampedSoftwareStartTrigger
```

## Inheritance

[object](#) ← TimestampedSoftwareStartTrigger

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [SoftwareStartTrigger](#) register. This field is constant.

```
public const int Address = 39
```

## Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [SoftwareStartTrigger](#) register messages.

```
public static Timestamped<byte> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[byte](#)>

A value representing the timestamped message payload.

## See Also

[SoftwareStartTrigger](#)

# Class TimestampedSoftwareStopTrigger

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Provides methods for manipulating timestamped messages from the SoftwareStopTrigger register.

```
public class TimestampedSoftwareStopTrigger
```

## Inheritance

[object](#) ← TimestampedSoftwareStopTrigger

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [SoftwareStopTrigger](#) register. This field is constant.

```
public const int Address = 40
```

## Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [SoftwareStopTrigger](#) register messages.

```
public static Timestamped<byte> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[byte](#)>

A value representing the timestamped message payload.

## See Also

[SoftwareStopTrigger](#)

# Class TimestampedTaskControl

Namespace: [AllenNeuralDynamics.Cuttlefish](#)

Assembly: AllenNeuralDynamics.Cuttlefish.dll

Provides methods for manipulating timestamped messages from the TaskControl register.

```
public class TimestampedTaskControl
```

## Inheritance

[object](#) ← TimestampedTaskControl

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [TaskControl](#) register. This field is constant.

```
public const int Address = 41
```

## Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [TaskControl](#) register messages.

```
public static Timestamped<TaskControlPayload> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[TaskControlPayload](#)>

A value representing the timestamped message payload.

## See Also

[TaskControl](#)

# Namespace AllenNeuralDynamics.EnvironmentSensor

## Classes

### [AsyncDevice](#)

Represents an asynchronous API to configure and interface with EnvironmentSensor devices.

### [CreateEnableEventsPayload](#)

Represents an operator that creates a message payload that enables (~2Hz) or disables the SensorData events.

### [CreateHumidityPayload](#)

Represents an operator that creates a message payload that humidity, in %RH.

### [CreateMessage](#)

Represents an operator which creates standard message payloads for the EnvironmentSensor device.

### [CreatePressurePayload](#)

Represents an operator that creates a message payload that pressure, in Pa.

### [CreateSensorDataPayload](#)

Represents an operator that creates a message payload that a periodic event will be emitted with aggregated data from all sensors.

### [CreateTemperaturePayload](#)

Represents an operator that creates a message payload that temperature in degrees C.

### [CreateTimestampedEnableEventsPayload](#)

Represents an operator that creates a timestamped message payload that enables (~2Hz) or disables the SensorData events.

### [CreateTimestampedHumidityPayload](#)

Represents an operator that creates a timestamped message payload that humidity, in %RH.

### [CreateTimestampedPressurePayload](#)

Represents an operator that creates a timestamped message payload that pressure, in Pa.

### [CreateTimestampedSensorDataPayload](#)

Represents an operator that creates a timestamped message payload that a periodic event will be emitted with aggregated data from all sensors.

### [CreateTimestampedTemperaturePayload](#)

Represents an operator that creates a timestamped message payload that temperature in degrees C.

## [Device](#)

Represents an observable source of messages from the Harp device connected at the specified serial port.

## [EnableEvents](#)

Represents a register that enables (~2Hz) or disables the SensorData events.

## [FilterRegister](#)

Represents an operator that filters register-specific messages reported by the [AllenNeuralDynamics.EnvironmentSensor](#) device.

## [Format](#)

Represents an operator which formats a sequence of values as specific EnvironmentSensor register messages.

## [GetMetadata](#)

Represents an operator that returns the contents of the metadata file describing the [AllenNeuralDynamics.EnvironmentSensor](#) device registers.

## [GroupByRegister](#)

Represents an operator that groups the sequence of [AllenNeuralDynamics.EnvironmentSensor](#)" messages by register type.

## [Humidity](#)

Represents a register that humidity, in %RH.

## [Parse](#)

Represents an operator which filters and selects specific messages reported by the EnvironmentSensor device.

## [Pressure](#)

Represents a register that pressure, in Pa.

## [SensorData](#)

Represents a register that a periodic event will be emitted with aggregated data from all sensors.

## [Temperature](#)

Represents a register that temperature in degrees C.

## [TimestampedEnableEvents](#)

Provides methods for manipulating timestamped messages from the EnableEvents register.

## [TimestampedHumidity](#)

Provides methods for manipulating timestamped messages from the Humidity register.

## [TimestampedPressure](#)

Provides methods for manipulating timestamped messages from the Pressure register.

## [TimestampedSensorData](#)

Provides methods for manipulating timestamped messages from the SensorData register.

## [TimestampedTemperature](#)

Provides methods for manipulating timestamped messages from the Temperature register.

# Structs

## [SensorDataPayload](#)

Represents the payload of the SensorData register.

# Enums

## [Events](#)

Available events on the device

# Class AsyncDevice

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents an asynchronous API to configure and interface with EnvironmentSensor devices.

```
public class AsyncDevice : AsyncDevice, IDisposable
```

## Inheritance

[object](#) ← [AsyncDevice](#) ← AsyncDevice

## Implements

[IDisposable](#)

## Inherited Members

[AsyncDevice.ReadWhoAmIAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadHardwareVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadAssemblyVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadCoreVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadFirmwareVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadTimestampSecondsAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadTimestampMicrosecondsAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadOperationControlAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadResetDeviceAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadDeviceNameAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadSerialNumberAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadClockConfigurationAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadByteAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadByteArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSByteAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSByteArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt16Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt16ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt16Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt16ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt32Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt32ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt32Async\(int, CancellationToken\)](#) ,

[AsyncDevice.ReadInt32ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt64Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt64ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt64Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt64ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSingleAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSingleArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.WriteTimestampSecondsAsync\(uint, CancellationToken\)](#) ,  
[AsyncDevice.WriteResetDeviceAsync\(ResetFlags, CancellationToken\)](#) ,  
[AsyncDevice.WriteDeviceNameAsync\(string, CancellationToken\)](#) ,  
[AsyncDevice.WriteClockConfigurationAsync\(ClockConfigurationFlags, CancellationToken\)](#) ,  
[AsyncDevice.WriteByteAsync\(int, byte, CancellationToken\)](#) ,  
[AsyncDevice.WriteByteAsync\(int, byte\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteSByteAsync\(int, sbyte, CancellationToken\)](#) ,  
[AsyncDevice.WriteSByteAsync\(int, sbyte\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteUInt16Async\(int, ushort, CancellationToken\)](#) ,  
[AsyncDevice.WriteUInt16Async\(int, ushort\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine16Async\(int, short, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine16Async\(int, short\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, uint, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, uint\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, int, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, int\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, ulong, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, ulong\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, long, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, long\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteSingleAsync\(int, float, CancellationToken\)](#) ,  
[AsyncDevice.WriteSingleAsync\(int, float\[\], CancellationToken\)](#) ,  
[AsyncDevice.CommandAsync\(HarpMessage, CancellationToken\)](#) , [AsyncDevice.Dispose\(\)](#) ,  
[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Methods

### ReadEnableEventsAsync(CancellationToken)

Asynchronously reads the contents of the EnableEvents register.

```
public Task<Events> ReadEnableEventsAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Events](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadHumidityAsync(CancellationToken)

Asynchronously reads the contents of the Humidity register.

```
public Task<float> ReadHumidityAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[float](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadPressureAsync(CancellationToken)

Asynchronously reads the contents of the Pressure register.

```
public Task<uint> ReadPressureAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[uint](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadSensorDataAsync(CancellationToken)

Asynchronously reads the contents of the SensorData register.

```
public Task<SensorDataPayload> ReadSensorDataAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[SensorDataPayload](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadTemperatureAsync(CancellationToken)

Asynchronously reads the contents of the Temperature register.

```
public Task<float> ReadTemperatureAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken  [CancellationToken](#)

A  [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#) < [float](#)>

A task that represents the asynchronous read operation. The  [Result](#) property contains the register payload.

## ReadTimestampedEnableEventsAsync(CancellationToken)

Asynchronously reads the timestamped contents of the EnableEvents register.

```
public Task<Timestamped<Events>> ReadTimestampedEnableEventsAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken  [CancellationToken](#)

A  [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#) < [Timestamped](#)< [Events](#)>>

A task that represents the asynchronous read operation. The  [Result](#) property contains the timestamped register payload.

## ReadTimestampedHumidityAsync(CancellationToken)

Asynchronously reads the timestamped contents of the Humidity register.

```
public Task<Timestamped<float>> ReadTimestampedHumidityAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<float>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedPressureAsync(CancellationToken)

Asynchronously reads the timestamped contents of the Pressure register.

```
public Task<Timestamped<uint>> ReadTimestampedPressureAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<uint>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedSensorDataAsync(CancellationToken)

Asynchronously reads the timestamped contents of the SensorData register.

```
public Task<Timestamped<SensorDataPayload>> ReadTimestampedSensorDataAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<[SensorDataPayload](#)>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedTemperatureAsync(CancellationToken)

Asynchronously reads the timestamped contents of the Temperature register.

```
public Task<Timestamped<float>> ReadTimestampedTemperatureAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<[float](#)>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## WriteEnableEventsAsync(Events, CancellationToken)

Asynchronously writes a value to the EnableEvents register.

```
public Task WriteEnableEventsAsync(Events value, CancellationToken cancellationToken  
= default)
```

## Parameters

### value [Events](#)

The value to be stored in the register.

### cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

### [Task](#)

The task object representing the asynchronous write operation.

# Class CreateEnableEventsPayload

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents an operator that creates a message payload that enables (~2Hz) or disables the SensorData events.

```
public class CreateEnableEventsPayload
```

## Inheritance

[object](#) ← CreateEnableEventsPayload

## Derived

[CreateTimestampedEnableEventsPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## EnableEvents

Gets or sets the value that enables (~2Hz) or disables the SensorData events.

```
public Events EnableEvents { get; set; }
```

## Property Value

[Events](#)

# Methods

## GetMessage(MessageType)

Creates a message that enables (~2Hz) or disables the SensorData events.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the EnableEvents register.

## GetPayload()

Creates a message payload for the EnableEvents register.

```
public Events GetPayload()
```

## Returns

[Events](#)

The created message payload value.

# Class CreateHumidityPayload

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents an operator that creates a message payload that humidity, in %RH.

```
public class CreateHumidityPayload
```

## Inheritance

[object](#) ← CreateHumidityPayload

## Derived

[CreateTimestampedHumidityPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Humidity

Gets or sets the value that humidity, in %RH.

```
public float Humidity { get; set; }
```

## Property Value

[float](#)

# Methods

## GetMessage(MessageType)

Creates a message that humidity, in %RH.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the Humidity register.

## GetPayload()

Creates a message payload for the Humidity register.

```
public float GetPayload()
```

## Returns

[float](#)

The created message payload value.

# Class CreateMessage

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents an operator which creates standard message payloads for the EnvironmentSensor device.

```
public class CreateMessage : CreateMessageBuilder, IExpressionBuilder,  
ICustomTypeDescriptor, INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [CreateMessageBuilder](#) ← CreateMessage

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[CreateMessageBuilder.Build\(IEnumerable<Expression>\)](#), [CreateMessageBuilder.ArgumentRange](#),  
[CreateMessageBuilder.MessageType](#), [CreateMessageBuilder.Payload](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

## Constructors

### CreateMessage()

Initializes a new instance of the [CreateMessage](#) class.

```
public CreateMessage()
```

## See Also

[CreatePressurePayload](#)

[CreateTemperaturePayload](#)

[CreateHumidityPayload](#)

[CreateSensorDataPayload](#)

[CreateEnableEventsPayload](#)

# Class CreatePressurePayload

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents an operator that creates a message payload that pressure, in Pa.

```
public class CreatePressurePayload
```

## Inheritance

[object](#) ← CreatePressurePayload

## Derived

[CreateTimestampedPressurePayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Pressure

Gets or sets the value that pressure, in Pa.

```
public uint Pressure { get; set; }
```

## Property Value

[uint](#)

# Methods

## GetMessage(MessageType)

Creates a message that pressure, in Pa.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the Pressure register.

## GetPayload()

Creates a message payload for the Pressure register.

```
public uint GetPayload()
```

## Returns

[uint](#)

The created message payload value.

# Class CreateSensorDataPayload

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents an operator that creates a message payload that a periodic event will be emitted with aggregated data from all sensors.

```
public class CreateSensorDataPayload
```

## Inheritance

[object](#) ← CreateSensorDataPayload

## Derived

[CreateTimestampedSensorDataPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Humidity

Gets or sets a value that humidity, in %RH.

```
public float Humidity { get; set; }
```

## Property Value

[float](#)

## Pressure

Gets or sets a value that pressure, in Pa.

```
public float Pressure { get; set; }
```

Property Value

[float](#)

## Temperature

Gets or sets a value that temperature in degrees C.

```
public float Temperature { get; set; }
```

Property Value

[float](#)

## Methods

### GetMessage(MessageType)

Creates a message that a periodic event will be emitted with aggregated data from all sensors.

```
public HarpMessage GetMessage(MessageType messageType)
```

Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

Returns

[HarpMessage](#)

A new message for the SensorData register.

## GetPayload()

Creates a message payload for the SensorData register.

```
public SensorDataPayload GetPayload()
```

Returns

[SensorDataPayload](#)

The created message payload value.

# Class CreateTemperaturePayload

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents an operator that creates a message payload that temperature in degrees C.

```
public class CreateTemperaturePayload
```

## Inheritance

[object](#) ← CreateTemperaturePayload

## Derived

[CreateTimestampedTemperaturePayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Temperature

Gets or sets the value that temperature in degrees C.

```
public float Temperature { get; set; }
```

## Property Value

[float](#)

# Methods

## GetMessage(MessageType)

Creates a message that temperature in degrees C.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the Temperature register.

## GetPayload()

Creates a message payload for the Temperature register.

```
public float GetPayload()
```

## Returns

[float](#)

The created message payload value.

# Class CreateTimestampedEnableEventsPayload

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents an operator that creates a timestamped message payload that enables (~2Hz) or disables the SensorData events.

```
public class CreateTimestampedEnableEventsPayload : CreateEnableEventsPayload
```

## Inheritance

[object](#) ← [CreateEnableEventsPayload](#) ← CreateTimestampedEnableEventsPayload

## Inherited Members

[CreateEnableEventsPayload.EnableEvents](#) , [CreateEnableEventsPayload.GetPayload\(\)](#) ,  
[CreateEnableEventsPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that enables (~2Hz) or disables the SensorData events.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

#### Returns

## [HarpMessage](#)

A new timestamped message for the EnableEvents register.

# Class CreateTimestampedHumidityPayload

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents an operator that creates a timestamped message payload that humidity, in %RH.

```
public class CreateTimestampedHumidityPayload : CreateHumidityPayload
```

## Inheritance

[object](#) ← [CreateHumidityPayload](#) ← CreateTimestampedHumidityPayload

## Inherited Members

[CreateHumidityPayload.Humidity](#) , [CreateHumidityPayload.GetPayload\(\)](#) ,  
[CreateHumidityPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that humidity, in %RH.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

#### Returns

[HarpMessage](#)

A new timestamped message for the Humidity register.

# Class CreateTimestampedPressurePayload

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents an operator that creates a timestamped message payload that pressure, in Pa.

```
public class CreateTimestampedPressurePayload : CreatePressurePayload
```

## Inheritance

[object](#) ← [CreatePressurePayload](#) ← CreateTimestampedPressurePayload

## Inherited Members

[CreatePressurePayload.Pressure](#) , [CreatePressurePayload.GetPayload\(\)](#) ,  
[CreatePressurePayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that pressure, in Pa.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

#### Returns

[HarpMessage](#)

A new timestamped message for the Pressure register.

# Class CreateTimestampedSensorDataPayload

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents an operator that creates a timestamped message payload that a periodic event will be emitted with aggregated data from all sensors.

```
public class CreateTimestampedSensorDataPayload : CreateSensorDataPayload
```

## Inheritance

[object](#) ← [CreateSensorDataPayload](#) ← CreateTimestampedSensorDataPayload

## Inherited Members

[CreateSensorDataPayload.Pressure](#) , [CreateSensorDataPayload.Temperature](#) ,  
[CreateSensorDataPayload.Humidity](#) , [CreateSensorDataPayload.GetPayload\(\)](#) ,  
[CreateSensorDataPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that a periodic event will be emitted with aggregated data from all sensors.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new timestamped message for the SensorData register.

# Class CreateTimestampedTemperaturePayload

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents an operator that creates a timestamped message payload that temperature in degrees C.

```
public class CreateTimestampedTemperaturePayload : CreateTemperaturePayload
```

## Inheritance

[object](#) ← [CreateTemperaturePayload](#) ← CreateTimestampedTemperaturePayload

## Inherited Members

[CreateTemperaturePayload.Temperature](#) , [CreateTemperaturePayload.GetPayload\(\)](#) ,  
[CreateTemperaturePayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that temperature in degrees C.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

#### Returns

[HarpMessage](#)

A new timestamped message for the Temperature register.

# Class Device

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents an observable source of messages from the Harp device connected at the specified serial port.

```
[Combinator(MethodName = "Generate")]
[WorkflowElementCategory(ElementCategory.Source)]
public class Device : Device, INamedElement
```

## Inheritance

[object](#) ← [Source](#)<[HarpMessage](#)> ← [Device](#) ← [Device](#)

## Implements

[INamedElement](#)

## Inherited Members

[Device.Generate\(\)](#), [Device.Generate\(IObservable<HarpMessage>\)](#), [Device.OperationMode](#),  
[Device.OperationLed](#), [Device.DumpRegisters](#), [Device.VisualIndicators](#), [Device.Heartbeat](#),  
[Device.IgnoreErrors](#), [Device.PortName](#), [object.ToString\(\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

# Constructors

## Device()

Initializes a new instance of the [Device](#) class.

```
public Device()
```

# Fields

## Metadata

Gets the contents of the metadata file describing the [AllenNeuralDynamics.EnvironmentSensor](#) device registers.

```
public static readonly string Metadata
```

Field Value

[string](#)

## WhoAmI

Represents the unique identity class of the [AllenNeuralDynamics.EnvironmentSensor](#) device. This field is constant.

```
public const int WhoAmI = 1405
```

Field Value

[int](#)

## Properties

### RegisterMap

Gets a read-only mapping from address to register type.

```
public static IReadOnlyDictionary<int, Type> RegisterMap { get; }
```

Property Value

[IReadOnlyDictionary](#)<[int](#), [Type](#)>

## Methods

### CreateAsync(string)

Initializes a new instance of the asynchronous API to configure and interface with EnvironmentSensor devices on the specified serial port.

```
public static Task<AsyncDevice> CreateAsync(string portName)
```

## Parameters

**portName** [string](#)

The name of the serial port used to communicate with the Harp device.

## Returns

[Task](#) <[AsyncDevice](#)>

A task that represents the asynchronous initialization operation. The value of the [Result](#) parameter contains a new instance of the [AsyncDevice](#) class.

# Class EnableEvents

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents a register that enables (~2Hz) or disables the SensorData events.

```
public class EnableEvents
```

## Inheritance

[object](#) ← EnableEvents

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [EnableEvents](#) register. This field is constant.

```
public const int Address = 36
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [EnableEvents](#) register. This field is constant.

```
public const int RegisterLength = 1
```

### Field Value

## .RegisterType

Represents the payload type of the [EnableEvents](#) register. This field is constant.

```
public const PayloadType RegisterType = U8
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, Events)

Returns a Harp message for the [EnableEvents](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, Events value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [Events](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [EnableEvents](#) register with the specified message type and payload.

### FromPayload(double, MessageType, Events)

Returns a timestamped Harp message for the [EnableEvents](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
Events value)
```

Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [Events](#)

The value to be stored in the message payload.

Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [EnableEvents](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [EnableEvents](#) register messages.

```
public static Events GetPayload(HarpMessage message)
```

Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

Returns

[Events](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [EnableEvents](#) register messages.

```
public static Timestamped<Events> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <[Events](#)>

A value representing the timestamped message payload.

# Enum Events

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Available events on the device

```
[Flags]  
public enum Events : byte
```

## Fields

Disable = 0

SensorData = 1

# Class FilterRegister

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents an operator that filters register-specific messages reported by the [AllenNeuralDynamics.EnvironmentSensor](#) device.

```
public class FilterRegister : FilterRegisterBuilder, IExpressionBuilder,  
ICustomTypeDescriptor, INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [RegisterCombinatorBuilder](#) ←  
[FilterRegisterBuilder](#) ← FilterRegister

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[FilterRegisterBuilder.FilterType](#), [FilterRegisterBuilder.Register](#),  
[RegisterCombinatorBuilder.Build\(IEnumerable<Expression>\)](#),  
[RegisterCombinatorBuilder.ArgumentRange](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

## Constructors

### FilterRegister()

Initializes a new instance of the [FilterRegister](#) class.

```
public FilterRegister()
```

## See Also

[Pressure](#)

[Temperature](#)

[Humidity](#)

[SensorData](#)

[EnableEvents](#)

# Class Format

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents an operator which formats a sequence of values as specific EnvironmentSensor register messages.

```
public class Format : FormatBuilder, IExpressionBuilder, ICustomTypeDescriptor,  
INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [FormatBuilder](#) ← Format

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[FormatBuilder.Build\(IEnumerable<Expression>\)](#), [FormatBuilder.ArgumentRange](#),  
[FormatBuilder.MessageType](#), [FormatBuilder.Register](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

# Constructors

## Format()

Initializes a new instance of the [Format](#) class.

```
public Format()
```

## See Also

[Pressure](#)

[Temperature](#)

[Humidity](#)

[SensorData](#)

[EnableEvents](#)

# Class GetMetadata

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents an operator that returns the contents of the metadata file describing the [AllenNeuralDynamics.EnvironmentSensor](#) device registers.

```
public class GetMetadata : Source<string>
```

## Inheritance

[object](#) ← [Source](#)<[string](#)> ← GetMetadata

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Methods

### Generate()

Returns an observable sequence with the contents of the metadata file describing the [AllenNeuralDynamics.EnvironmentSensor](#) device registers.

```
public override IObservable<string> Generate()
```

## Returns

[IObservable](#)<[string](#)>

A sequence with a single [string](#) object representing the contents of the metadata file.

# Class GroupByRegister

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents an operator that groups the sequence of [AllenNeuralDynamics.EnvironmentSensor](#)" messages by register type.

```
public class GroupByRegister : Combinator<HarpMessage, IGroupedObservable<Type, HarpMessage>>
```

## Inheritance

```
object ↗ ← Combinator ↗ <HarpMessage ↗, IGroupedObservable ↗ <Type ↗, HarpMessage ↗>> ← GroupByRegister
```

## Inherited Members

```
Combinator<HarpMessage, IGroupedObservable<Type, HarpMessage>>.Process(I Observable<HarpMessage>) ↗ ,  
object.ToString() ↗ , object.Equals(object) ↗ , object.Equals(object, object) ↗ ,  
object.ReferenceEquals(object, object) ↗ , object.GetHashCode() ↗ , object.GetType() ↗ ,  
object.MemberwiseClone() ↗
```

## Methods

### Process(I Observable<HarpMessage>)

Groups an observable sequence of [AllenNeuralDynamics.EnvironmentSensor](#) messages by register type.

```
public override IObservable<IGroupedObservable<Type, HarpMessage>>  
Process(IObservable<HarpMessage> source)
```

## Parameters

source [IObservable](#)<HarpMessage>

The sequence of Harp device messages.

## Returns

[IObservable](#) < [IGroupedObservable](#) < [Type](#), [HarpMessage](#) >>

A sequence of observable groups, each of which corresponds to a unique [AllenNeuralDynamics.EnvironmentSensor](#) register.

# Class Humidity

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents a register that humidity, in %RH.

```
public class Humidity
```

## Inheritance

[object](#) ← Humidity

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [Humidity](#) register. This field is constant.

```
public const int Address = 34
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [Humidity](#) register. This field is constant.

```
public const int RegisterLength = 1
```

### Field Value

## .RegisterType

Represents the payload type of the [Humidity](#) register. This field is constant.

```
public const PayloadType RegisterType = Float
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, float)

Returns a Harp message for the [Humidity](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, float value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [float](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [Humidity](#) register with the specified message type and payload.

### FromPayload(double, MessageType, float)

Returns a timestamped Harp message for the [Humidity](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
float value)
```

Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [float](#)

The value to be stored in the message payload.

Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [Humidity](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [Humidity](#) register messages.

```
public static float GetPayload(HarpMessage message)
```

Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

Returns

[float](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [Humidity](#) register messages.

```
public static Timestamped<float> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <[float](#)>

A value representing the timestamped message payload.

# Class Parse

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents an operator which filters and selects specific messages reported by the EnvironmentSensor device.

```
public class Parse : ParseBuilder, IExpressionBuilder, ICustomTypeDescriptor, INamedElement
```

## Inheritance

```
object ↵ ← ExpressionBuilder ↵ ← HarpCombinatorBuilder ↵ ← RegisterCombinatorBuilder ↵ ←  
ParseBuilder ↵ ← Parse
```

## Implements

```
IExpressionBuilder ↵ , ICustomTypeDescriptor ↵ , INamedElement ↵
```

## Inherited Members

```
ParseBuilder.Register() , RegisterCombinatorBuilder.Build(IEnumerable<Expression>) ↵ ,  
RegisterCombinatorBuilder.ArgumentRange() , ExpressionBuilder.ToString() ↵ ,  
ExpressionBuilder.Unwrap(ExpressionBuilder) ↵ ,  
ExpressionBuilder.GetWorkflowElement(ExpressionBuilder) ↵ ,  
ExpressionBuilder.GetVisualizerElement(ExpressionBuilder) ↵ ,  
ExpressionBuilder.GetVisualizerMappings(ExpressionBuilder) ↵ ,  
ExpressionBuilder.FromWorkflowElement(object, ElementCategory) ↵ ,  
ExpressionBuilder.GetElementDisplayName(Type) ↵ ,  
ExpressionBuilder.GetElementDisplayName(object) ↵ , object.Equals(object) ↵ ,  
object.Equals(object, object) ↵ , object.ReferenceEquals(object, object) ↵ , object.GetHashCode() ↵ ,  
object.GetType() ↵ , object.MemberwiseClone() ↵
```

# Constructors

## Parse()

Initializes a new instance of the [Parse](#) class.

```
public Parse()
```

## See Also

[Pressure](#)

[Temperature](#)

[Humidity](#)

[SensorData](#)

[EnableEvents](#)

# Class Pressure

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents a register that pressure, in Pa.

```
public class Pressure
```

## Inheritance

[object](#) ← Pressure

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [Pressure](#) register. This field is constant.

```
public const int Address = 32
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [Pressure](#) register. This field is constant.

```
public const int RegisterLength = 1
```

### Field Value

## .RegisterType

Represents the payload type of the [Pressure](#) register. This field is constant.

```
public const PayloadType RegisterType = U32
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, uint)

Returns a Harp message for the [Pressure](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, uint value)
```

#### Parameters

`messageType` [MessageType](#)

The type of the Harp message.

`value` [uint](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [Pressure](#) register with the specified message type and payload.

### FromPayload(double, MessageType, uint)

Returns a timestamped Harp message for the [Pressure](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType, uint value)
```

## Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [uint](#)

The value to be stored in the message payload.

## Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [Pressure](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [Pressure](#) register messages.

```
public static uint GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[uint](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [Pressure](#) register messages.

```
public static Timestamped<uint> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#)<[uint](#)>

A value representing the timestamped message payload.

# Class SensorData

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents a register that a periodic event will be emitted with aggregated data from all sensors.

```
public class SensorData
```

## Inheritance

[object](#) ← SensorData

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [SensorData](#) register. This field is constant.

```
public const int Address = 35
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [SensorData](#) register. This field is constant.

```
public const int RegisterLength = 3
```

### Field Value

## .RegisterType

Represents the payload type of the [SensorData](#) register. This field is constant.

```
public const PayloadType RegisterType = Float
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, SensorDataPayload)

Returns a Harp message for the [SensorData](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, SensorDataPayload value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [SensorDataPayload](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [SensorData](#) register with the specified message type and payload.

### FromPayload(double, MessageType, SensorDataPayload)

Returns a timestamped Harp message for the [SensorData](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
SensorDataPayload value)
```

Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [SensorDataPayload](#)

The value to be stored in the message payload.

Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [SensorData](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [SensorData](#) register messages.

```
public static SensorDataPayload GetPayload(HarpMessage message)
```

Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

Returns

[SensorDataPayload](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [SensorData](#) register messages.

```
public static Timestamped<SensorDataPayload> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <[SensorDataPayload](#)>

A value representing the timestamped message payload.

# Struct SensorDataPayload

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents the payload of the SensorData register.

```
public struct SensorDataPayload
```

## Inherited Members

[ValueType.Equals\(object\)](#) , [ValueType.GetHashCode\(\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetType\(\)](#)

## Constructors

### SensorDataPayload(float, float, float)

Initializes a new instance of the [SensorDataPayload](#) structure.

```
public SensorDataPayload(float pressure, float temperature, float humidity)
```

#### Parameters

**pressure** [float](#)

Pressure, in Pa

**temperature** [float](#)

Temperature in degrees C

**humidity** [float](#)

Humidity, in %RH

## Fields

## Humidity

Humidity, in %RH

```
public float Humidity
```

Field Value

[float](#)

## Pressure

Pressure, in Pa

```
public float Pressure
```

Field Value

[float](#)

## Temperature

Temperature in degrees C

```
public float Temperature
```

Field Value

[float](#)

## Methods

### ToString()

Returns a [string](#) that represents the payload of the SensorData register.

```
public override string ToString()
```

Returns

[string](#)

A [string](#) that represents the payload of the SensorData register.

# Class Temperature

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Represents a register that temperature in degrees C.

```
public class Temperature
```

## Inheritance

[object](#) ← Temperature

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [Temperature](#) register. This field is constant.

```
public const int Address = 33
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [Temperature](#) register. This field is constant.

```
public const int RegisterLength = 1
```

### Field Value

## .RegisterType

Represents the payload type of the [Temperature](#) register. This field is constant.

```
public const PayloadType RegisterType = Float
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, float)

Returns a Harp message for the [Temperature](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, float value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [float](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [Temperature](#) register with the specified message type and payload.

### FromPayload(double, MessageType, float)

Returns a timestamped Harp message for the [Temperature](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
float value)
```

Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [float](#)

The value to be stored in the message payload.

Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [Temperature](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [Temperature](#) register messages.

```
public static float GetPayload(HarpMessage message)
```

Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

Returns

[float](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [Temperature](#) register messages.

```
public static Timestamped<float> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#)<[float](#)>

A value representing the timestamped message payload.

# Class TimestampedEnableEvents

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Provides methods for manipulating timestamped messages from the EnableEvents register.

```
public class TimestampedEnableEvents
```

## Inheritance

[object](#) ← TimestampedEnableEvents

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [EnableEvents](#) register. This field is constant.

```
public const int Address = 36
```

## Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [EnableEvents](#) register messages.

```
public static Timestamped<Events> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[Events](#)>

A value representing the timestamped message payload.

## See Also

[EnableEvents](#)

# Class TimestampedHumidity

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Provides methods for manipulating timestamped messages from the Humidity register.

```
public class TimestampedHumidity
```

## Inheritance

[object](#) ← TimestampedHumidity

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [Humidity](#) register. This field is constant.

```
public const int Address = 34
```

## Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [Humidity](#) register messages.

```
public static Timestamped<float> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[float](#)>

A value representing the timestamped message payload.

## See Also

[Humidity](#)

# Class TimestampedPressure

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Provides methods for manipulating timestamped messages from the Pressure register.

```
public class TimestampedPressure
```

## Inheritance

[object](#) ← TimestampedPressure

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [Pressure](#) register. This field is constant.

```
public const int Address = 32
```

## Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [Pressure](#) register messages.

```
public static Timestamped<uint> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[uint](#)>

A value representing the timestamped message payload.

## See Also

[Pressure](#)

# Class TimestampedSensorData

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Provides methods for manipulating timestamped messages from the SensorData register.

```
public class TimestampedSensorData
```

## Inheritance

[object](#) ← TimestampedSensorData

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [SensorData](#) register. This field is constant.

```
public const int Address = 35
```

### Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [SensorData](#) register messages.

```
public static Timestamped<SensorDataPayload> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[SensorDataPayload](#)>

A value representing the timestamped message payload.

## See Also

[SensorData](#)

# Class TimestampedTemperature

Namespace: [AllenNeuralDynamics.EnvironmentSensor](#)

Assembly: AllenNeuralDynamics.EnvironmentSensor.dll

Provides methods for manipulating timestamped messages from the Temperature register.

```
public class TimestampedTemperature
```

## Inheritance

[object](#) ← TimestampedTemperature

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [Temperature](#) register. This field is constant.

```
public const int Address = 33
```

### Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [Temperature](#) register messages.

```
public static Timestamped<float> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[float](#)>

A value representing the timestamped message payload.

## See Also

[Temperature](#)

# Namespace AllenNeuralDynamics.HarpUtils

## Classes

[CreateHarpMetadata](#)

[CreateOdorMix](#)

[HarpDeviceMetadata](#)

[ModifyMessage](#)

[OdorMixMessages](#)

[ValidateClkOutputChannels](#)

[ValidateClkOutputChannelsDiagnosis](#)

[ValidatedDevice](#)

# Class CreateHarpMetadata

Namespace: [AllenNeuralDynamics.HarpUtils](#)

Assembly: AllenNeuralDynamics.HarpUtils.dll

```
public class CreateHarpMetadata : Source<HarpDeviceMetadata>
```

## Inheritance

[object](#) ← [Source](#) <[HarpDeviceMetadata](#)> ← CreateHarpMetadata

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Properties

### AssemblyVersion

```
public int? AssemblyVersion { get; set; }
```

#### Property Value

[int](#)?

### DeviceName

```
public string DeviceName { get; set; }
```

#### Property Value

[string](#)

### FirmwareTag

```
public int? FirmwareTag { get; set; }
```

Property Value

[int](#)?

## MajorCoreVersion

```
public int? MajorCoreVersion { get; set; }
```

Property Value

[int](#)?

## MajorFirmwareVersion

```
public int? MajorFirmwareVersion { get; set; }
```

Property Value

[int](#)?

## MajorHardwareVersion

```
public int? MajorHardwareVersion { get; set; }
```

Property Value

[int](#)?

## MinorCoreVersion

```
public int? MinorCoreVersion { get; set; }
```

Property Value

[int](#)?

## MinorFirmwareVersion

```
public int? MinorFirmwareVersion { get; set; }
```

Property Value

[int](#)?

## MinorHardwareVersion

```
public int? MinorHardwareVersion { get; set; }
```

Property Value

[int](#)?

## PrereleaseVersion

```
public int? PrereleaseVersion { get; set; }
```

Property Value

[int](#)?

## SerialNumber

```
public int? SerialNumber { get; set; }
```

Property Value

[int](#)?

WhoAmI

```
public int? WhoAmI { get; set; }
```

Property Value

[int](#)?

## Methods

Generate()

Generates an observable sequence of data elements.

```
public override IObservable<HarpDeviceMetadata> Generate()
```

Returns

[IObservable](#)<[HarpDeviceMetadata](#)>

An observable sequence of data elements of type [HarpDeviceMetadata](#).

Generate<TSource>(IObservable<TSource>)

```
public IObservable<HarpDeviceMetadata> Generate<TSource>(IObservable<TSource> source)
```

Parameters

source [IObservable](#)<TSource>

Returns

[IObservable](#) ↗ <[HarpDeviceMetadata](#)>

Type Parameters

**TSource**

# Class CreateOdorMix

Namespace: [AllenNeuralDynamics.HarpUtils](#)

Assembly: AllenNeuralDynamics.HarpUtils.dll

```
public class CreateOdorMix : Source<OdorMixMessages>
```

## Inheritance

[object](#) ← [Source](#)<[OdorMixMessages](#)> ← CreateOdorMix

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### channel3AsCarrier

```
public bool channel3AsCarrier
```

## Field Value

[bool](#)

## Properties

### Channel3AsCarrier

```
public bool Channel3AsCarrier { get; set; }
```

## Property Value

[bool](#)

## PercentageChannel0

```
[Range(0, 1)]  
public float PercentageChannel0 { get; set; }
```

Property Value

[float](#) ↗

## PercentageChannel1

```
[Range(0, 1)]  
public float PercentageChannel1 { get; set; }
```

Property Value

[float](#) ↗

## PercentageChannel2

```
[Range(0, 1)]  
public float PercentageChannel2 { get; set; }
```

Property Value

[float](#) ↗

## PercentageChannel3

```
[Range(0, 1)]  
public float PercentageChannel3 { get; set; }
```

Property Value

[float](#) ↗

## TargetOdorFlow

```
[Range(0, 100)]  
public int TargetOdorFlow { get; set; }
```

Property Value

[int ↗](#)

## TotalFlow

```
[Range(0, 1000)]  
public int TotalFlow { get; set; }
```

Property Value

[int ↗](#)

## Methods

### Generate()

Generates an observable sequence of data elements.

```
public override IObservable<OdorMixMessages> Generate()
```

Returns

[IObservable ↗ <OdorMixMessages>](#)

An observable sequence of data elements of type [OdorMixMessages](#).

### Generate(IObservable<Tuple<int, double>>)

```
public IObservable<OdorMixMessages> Generate(IObservable<Tuple<int, double>> source)
```

Parameters

source [IObservable<Tuple<int, double>>](#)

Returns

[IObservable<OdorMixMessages>](#)

## Generate<TSource>(IObservable<TSource>)

```
public IObservable<OdorMixMessages> Generate<TSource>(IObservable<TSource> source)
```

Parameters

source [IObservable<TSource>](#)

Returns

[IObservable<OdorMixMessages>](#)

Type Parameters

TSource

# Class HarpDeviceMetadata

Namespace: [AllenNeuralDynamics.HarpUtils](#)

Assembly: AllenNeuralDynamics.HarpUtils.dll

```
public class HarpDeviceMetadata
```

## Inheritance

[object](#) ← HarpDeviceMetadata

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## AssemblyVersion

```
public int? AssemblyVersion { get; set; }
```

### Property Value

[int](#)?

## DeviceName

```
public string DeviceName { get; set; }
```

### Property Value

[string](#)

## FirmwareTag

```
public int? FirmwareTag { get; set; }
```

Property Value

[int](#)?

## MajorCoreVersion

```
public int? MajorCoreVersion { get; set; }
```

Property Value

[int](#)?

## MajorFirmwareVersion

```
public int? MajorFirmwareVersion { get; set; }
```

Property Value

[int](#)?

## MajorHardwareVersion

```
public int? MajorHardwareVersion { get; set; }
```

Property Value

[int](#)?

## MinorCoreVersion

```
public int? MinorCoreVersion { get; set; }
```

Property Value

[int](#)?

## MinorFirmwareVersion

```
public int? MinorFirmwareVersion { get; set; }
```

Property Value

[int](#)?

## MinorHardwareVersion

```
public int? MinorHardwareVersion { get; set; }
```

Property Value

[int](#)?

## PrereleaseVersion

```
public int? PrereleaseVersion { get; set; }
```

Property Value

[int](#)?

## SerialNumber

```
public int? SerialNumber { get; set; }
```

Property Value

[int](#)?

## WhoAmI

```
public int? WhoAmI { get; set; }
```

Property Value

[int](#)?

## Methods

### Serialize()

```
public string Serialize()
```

Returns

[string](#)

### ToFirmwareMetadata()

```
public FirmwareMetadata ToFirmwareMetadata()
```

Returns

[FirmwareMetadata](#)

# Class ModifyMessage

Namespace: [AllenNeuralDynamics.HarpUtils](#)

Assembly: AllenNeuralDynamics.HarpUtils.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Transform)]
public class ModifyMessage
```

## Inheritance

[object](#) ← ModifyMessage

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Properties

### Address

Gets or sets the address of the register to which the Harp message refers to.

```
public int? Address { get; set; }
```

### Property Value

[int](#)?

### MessageTypeModified

Gets or sets the type of the Harp message.

```
public MessageType? MessageTypeModified { get; set; }
```

### Property Value

[MessageType](#)?

## Methods

### Process(I`Observable`<HarpMessage>)

```
public Ibservable<HarpMessage> Process(Ibservable<HarpMessage> source)
```

#### Parameters

source [I`bservable`](#)<[HarpMessage](#)>

#### Returns

[I`bservable`](#)<[HarpMessage](#)>

### Process(I`bservable`<Tuple<HarpMessage, double>>)

```
public Ibservable<HarpMessage> Process(Ibservable<Tuple<HarpMessage, double>> source)
```

#### Parameters

source [I`bservable`](#)<[Tuple](#)<[HarpMessage](#), [double](#)>>

#### Returns

[I`bservable`](#)<[HarpMessage](#)>

# Class OdorMixMessages

Namespace: [AllenNeuralDynamics.HarpUtils](#)

Assembly: AllenNeuralDynamics.HarpUtils.dll

```
public class OdorMixMessages
```

## Inheritance

[object](#) ← OdorMixMessages

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Properties

### ChannelsTargetFlow

```
public HarpMessage ChannelsTargetFlow { get; set; }
```

Property Value

[HarpMessage](#)

### OdorValveState

```
public HarpMessage OdorValveState { get; set; }
```

Property Value

[HarpMessage](#)

# Class ValidateClkOutputChannels

Namespace: [AllenNeuralDynamics.HarpUtils](#)

Assembly: AllenNeuralDynamics.HarpUtils.dll

```
[Combinator]
[WorkflowElementCategory(ElementCategory.Transform)]
public class ValidateClkOutputChannels
```

## Inheritance

[object](#) ← ValidateClkOutputChannels

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Properties

### ExpectedChannels

```
public Dictionary<int, string> ExpectedChannels { get; set; }
```

## Property Value

[Dictionary](#)<[int](#), [string](#)>

## Methods

### Process(IObservable<int>)

```
public IObservable<ValidateClkOutputChannelsDiagnosis> Process(IObservable<int> source)
```

## Parameters

source [IObservable<int>](#)

Returns

[IObservable<ValidateClkOutputChannelsDiagnosis>](#)

## Process<T>(IObservable<T>)

```
public IObservable<ValidateClkOutputChannelsDiagnosis> Process<T>(IObservable<T> source)
where T : struct, IConvertible
```

Parameters

source [IObservable<T>](#)

Returns

[IObservable<ValidateClkOutputChannelsDiagnosis>](#)

## Type Parameters

T

# Class ValidateClkOutputChannelsDiagnosis

Namespace: [AllenNeuralDynamics.HarpUtils](#)

Assembly: AllenNeuralDynamics.HarpUtils.dll

```
public class ValidateClkOutputChannelsDiagnosis
```

## Inheritance

[object](#) ← ValidateClkOutputChannelsDiagnosis

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Fields

### ExtraChannels

```
public int[] ExtraChannels
```

#### Field Value

[int](#)[]

### FoundChannels

```
public Dictionary<int, string> FoundChannels
```

#### Field Value

[Dictionary](#)<[int](#), [string](#)>

### MissingChannels

```
public Dictionary<int, string> MissingChannels
```

## Field Value

[Dictionary](#)<[int](#), [string](#)>

## Methods

### ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

## Returns

[string](#)

A string that represents the current object.

# Class ValidatedDevice

Namespace: [AllenNeuralDynamics.HarpUtils](#)

Assembly: AllenNeuralDynamics.HarpUtils.dll

```
[Combinator(MethodName = "Generate")]
[WorkflowElementCategory(ElementCategory.Source)]
public class ValidatedDevice
```

## Inheritance

[object](#) ← ValidatedDevice

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

# Properties

## DumpRegisters

```
public bool DumpRegisters { get; set; }
```

## Property Value

[bool](#)

## HarpMetadata

```
public HarpDeviceMetadata HarpMetadata { get; set; }
```

## Property Value

[HarpDeviceMetadata](#)

## IgnoreErrors

```
public bool IgnoreErrors { get; set; }
```

Property Value

[bool](#)

## OperationLed

```
public LedState OperationLed { get; set; }
```

Property Value

[LedState](#)

## OperationMode

```
public OperationMode OperationMode { get; set; }
```

Property Value

[OperationMode](#)

## PortName

```
[TypeConverter(typeof(PortNameConverter))]  
public string PortName { get; set; }
```

Property Value

[string](#)

## VisualIndicators

```
public LedState VisualIndicators { get; set; }
```

Property Value

[LedState](#)

## WhoAmI

```
public int WhoAmI { get; set; }
```

Property Value

[int](#)

## Methods

### Generate()

```
public IObservable<HarpMessage> Generate()
```

Returns

[IObservable](#) <[HarpMessage](#)>

### Generate(IObservable<HarpMessage>)

```
public IObservable<HarpMessage> Generate(IObservable<HarpMessage> source)
```

Parameters

source [IObservable](#) <[HarpMessage](#)>

Returns

[IObservable](#) <[HarpMessage](#)>

# Namespace AllenNeuralDynamics.LicketySplit

## Classes

### [AsyncDevice](#)

Represents an asynchronous API to configure and interface with LicketySplit devices.

### [Channel0TriggerThreshold](#)

Represents a register that threshold value to detect the lick. Values below this threshold will be considered a detected lick.

### [Channel0UntriggerThreshold](#)

Represents a register that threshold value to release the lick detection state. Values above this threshold will untrigger a detected lick.

### [CreateChannel0TriggerThresholdPayload](#)

Represents an operator that creates a message payload that threshold value to detect the lick. Values below this threshold will be considered a detected lick.

### [CreateChannel0UntriggerThresholdPayload](#)

Represents an operator that creates a message payload that threshold value to release the lick detection state. Values above this threshold will untrigger a detected lick.

### [CreateLickStatePayload](#)

Represents an operator that creates a message payload that emits an event when the state of any lick detector changes. Value will be High when lick detected and Low otherwise.

### [CreateMessage](#)

Represents an operator which creates standard message payloads for the LicketySplit device.

### [CreateTimestampedChannel0TriggerThresholdPayload](#)

Represents an operator that creates a timestamped message payload that threshold value to detect the lick. Values below this threshold will be considered a detected lick.

### [CreateTimestampedChannel0UntriggerThresholdPayload](#)

Represents an operator that creates a timestamped message payload that threshold value to release the lick detection state. Values above this threshold will untrigger a detected lick.

### [CreateTimestampedLickStatePayload](#)

Represents an operator that creates a timestamped message payload that emits an event when the state of any lick detector changes. Value will be High when lick detected and Low otherwise.

### [Device](#)

Represents an observable source of messages from the Harp device connected at the specified serial port.

### [FilterRegister](#)

Represents an operator that filters register-specific messages reported by the [AllenNeuralDynamics.LicketySplit](#) device.

### [Format](#)

Represents an operator which formats a sequence of values as specific LicketySplit register messages.

### [GetMetadata](#)

Represents an operator that returns the contents of the metadata file describing the [AllenNeuralDynamics.LicketySplit](#) device registers.

### [GroupByRegister](#)

Represents an operator that groups the sequence of [AllenNeuralDynamics.LicketySplit](#)" messages by register type.

### [LickState](#)

Represents a register that emits an event when the state of any lick detector changes. Value will be High when lick detected and Low otherwise.

### [Parse](#)

Represents an operator which filters and selects specific messages reported by the LicketySplit device.

### [TimestampedChannel0TriggerThreshold](#)

Provides methods for manipulating timestamped messages from the Channel0TriggerThreshold register.

### [TimestampedChannel0UntriggerThreshold](#)

Provides methods for manipulating timestamped messages from the Channel0UntriggerThreshold register.

### [TimestampedLickState](#)

Provides methods for manipulating timestamped messages from the LickState register.

## Enums

### [LickChannels](#)

The channel of the lick detector.

# Class AsyncDevice

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

Represents an asynchronous API to configure and interface with LicketySplit devices.

```
public class AsyncDevice : AsyncDevice, IDisposable
```

## Inheritance

[object](#) ← [AsyncDevice](#) ← AsyncDevice

## Implements

[IDisposable](#)

## Inherited Members

[AsyncDevice.ReadWhoAmIAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadHardwareVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadAssemblyVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadCoreVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadFirmwareVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadTimestampSecondsAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadTimestampMicrosecondsAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadOperationControlAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadResetDeviceAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadDeviceNameAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadSerialNumberAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadClockConfigurationAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadByteAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadByteArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSByteAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSByteArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt16Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt16ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt16Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt16ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt32Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt32ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt32Async\(int, CancellationToken\)](#) ,

[AsyncDevice.ReadInt32ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt64Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt64ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt64Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt64ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSingleAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSingleArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.WriteTimestampSecondsAsync\(uint, CancellationToken\)](#) ,  
[AsyncDevice.WriteResetDeviceAsync\(ResetFlags, CancellationToken\)](#) ,  
[AsyncDevice.WriteDeviceNameAsync\(string, CancellationToken\)](#) ,  
[AsyncDevice.WriteClockConfigurationAsync\(ClockConfigurationFlags, CancellationToken\)](#) ,  
[AsyncDevice.WriteByteAsync\(int, byte, CancellationToken\)](#) ,  
[AsyncDevice.WriteByteAsync\(int, byte\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteSByteAsync\(int, sbyte, CancellationToken\)](#) ,  
[AsyncDevice.WriteSByteAsync\(int, sbyte\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteUInt16Async\(int, ushort, CancellationToken\)](#) ,  
[AsyncDevice.WriteUInt16Async\(int, ushort\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine16Async\(int, short, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine16Async\(int, short\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, uint, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, uint\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, int, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, int\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, ulong, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, ulong\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, long, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, long\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteSingleAsync\(int, float, CancellationToken\)](#) ,  
[AsyncDevice.WriteSingleAsync\(int, float\[\], CancellationToken\)](#) ,  
[AsyncDevice.CommandAsync\(HarpMessage, CancellationToken\)](#) , [AsyncDevice.Dispose\(\)](#) ,  
[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Methods

### ReadChannel0TriggerThresholdAsync(CancellationToken)

Asynchronously reads the contents of the Channel0TriggerThreshold register.

```
public Task<byte> ReadChannel0TriggerThresholdAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#) <byte>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadChannel0UntriggerThresholdAsync(CancellationToken)

Asynchronously reads the contents of the Channel0UntriggerThreshold register.

```
public Task<byte> ReadChannel0UntriggerThresholdAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#) <byte>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadLickStateAsync(CancellationToken)

Asynchronously reads the contents of the LickState register.

```
public Task<LickChannels> ReadLickStateAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[LickChannels](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadTimestampedChannel0TriggerThresholdAsync(CancellationToken Token)

Asynchronously reads the timestamped contents of the Channel0TriggerThreshold register.

```
public Task<Timestamped<byte>>
ReadTimestampedChannel0TriggerThresholdAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<[byte](#)>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedChannel0UntriggerThresholdAsync(CancellationTokenToken)

Asynchronously reads the timestamped contents of the Channel0UntriggerThreshold register.

```
public Task<Timestamped<byte>>
ReadTimestampedChannel0UntriggerThresholdAsync(CancellationToken cancellationToken
= default)
```

## Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<[byte](#)>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedLickStateAsync(CancellationToken)

Asynchronously reads the timestamped contents of the LickState register.

```
public Task<Timestamped<LickChannels>> ReadTimestampedLickStateAsync(CancellationToken cancellationToken
= default)
```

## Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<[LickChannels](#)>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## WriteChannel0TriggerThresholdAsync(byte, CancellationToken)

Asynchronously writes a value to the Channel0TriggerThreshold register.

```
public Task WriteChannel0TriggerThresholdAsync(byte value, CancellationToken  
cancellationToken = default)
```

### Parameters

**value** [byte](#)

The value to be stored in the register.

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

### Returns

[Task](#)

The task object representing the asynchronous write operation.

## WriteChannel0UntriggerThresholdAsync(byte, CancellationToken)

Asynchronously writes a value to the Channel0UntriggerThreshold register.

```
public Task WriteChannel0UntriggerThresholdAsync(byte value, CancellationToken  
cancellationToken = default)
```

### Parameters

**value** [byte](#)

The value to be stored in the register.

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

### [Task](#) ↗

The task object representing the asynchronous write operation.

# Class Channel0TriggerThreshold

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

Represents a register that threshold value to detect the lick. Values below this threshold will be considered a detected lick.

```
public class Channel0TriggerThreshold
```

## Inheritance

[object](#) ← Channel0TriggerThreshold

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [Channel0TriggerThreshold](#) register. This field is constant.

```
public const int Address = 33
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [Channel0TriggerThreshold](#) register. This field is constant.

```
public const int RegisterLength = 1
```

## Field Value

[int↗](#)

## RegisterType

Represents the payload type of the [Channel0TriggerThreshold](#) register. This field is constant.

```
public const PayloadType RegisterType = U8
```

## Field Value

[PayloadType↗](#)

## Methods

### FromPayload(MessageType, byte)

Returns a Harp message for the [Channel0TriggerThreshold](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, byte value)
```

#### Parameters

**messageType** [MessageType↗](#)

The type of the Harp message.

**value** [byte↗](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage↗](#)

A [HarpMessage↗](#) object for the [Channel0TriggerThreshold](#) register with the specified message type and payload.

## FromPayload(double, MessageType, byte)

Returns a timestamped Harp message for the [Channel0TriggerThreshold](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType, byte value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [byte](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [Channel0TriggerThreshold](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [Channel0TriggerThreshold](#) register messages.

```
public static byte GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## [byte](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [Channel0TriggerThreshold](#) register messages.

```
public static Timestamped<byte> GetTimestampedPayload(HarpMessage message)
```

### Parameters

[message](#) [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <[byte](#)>

A value representing the timestamped message payload.

# Class Channel0UntriggerThreshold

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

Represents a register that threshold value to release the lick detection state. Values above this threshold will untrigger a detected lick.

```
public class Channel0UntriggerThreshold
```

## Inheritance

[object](#) ← Channel0UntriggerThreshold

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [Channel0UntriggerThreshold](#) register. This field is constant.

```
public const int Address = 34
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [Channel0UntriggerThreshold](#) register. This field is constant.

```
public const int RegisterLength = 1
```

## Field Value

[int↗](#)

## RegisterType

Represents the payload type of the [Channel0UntriggerThreshold](#) register. This field is constant.

```
public const PayloadType RegisterType = U8
```

## Field Value

[PayloadType↗](#)

## Methods

### FromPayload(MessageType, byte)

Returns a Harp message for the [Channel0UntriggerThreshold](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, byte value)
```

#### Parameters

**messageType** [MessageType↗](#)

The type of the Harp message.

**value** [byte↗](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage↗](#)

A [HarpMessage↗](#) object for the [Channel0UntriggerThreshold](#) register with the specified message type and payload.

## FromPayload(double, MessageType, byte)

Returns a timestamped Harp message for the [Channel0UntriggerThreshold](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType, byte value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [byte](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [Channel0UntriggerThreshold](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [Channel0UntriggerThreshold](#) register messages.

```
public static byte GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## [byte](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [Channel0UntriggerThreshold](#) register messages.

```
public static Timestamped<byte> GetTimestampedPayload(HarpMessage message)
```

### Parameters

[message](#) [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <[byte](#)>

A value representing the timestamped message payload.

# Class CreateChannel0TriggerThresholdPayload

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

Represents an operator that creates a message payload that threshold value to detect the lick. Values below this threshold will be considered a detected lick.

```
public class CreateChannel0TriggerThresholdPayload
```

## Inheritance

[object](#) ← CreateChannel0TriggerThresholdPayload

## Derived

[CreateTimestampedChannel0TriggerThresholdPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Channel0TriggerThreshold

Gets or sets the value that threshold value to detect the lick. Values below this threshold will be considered a detected lick.

```
public byte Channel0TriggerThreshold { get; set; }
```

## Property Value

[byte](#)

# Methods

## GetMessage(MessageType)

Creates a message that threshold value to detect the lick. Values below this threshold will be considered a detected lick.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the Channel0TriggerThreshold register.

## GetPayload()

Creates a message payload for the Channel0TriggerThreshold register.

```
public byte GetPayload()
```

## Returns

[byte](#)

The created message payload value.

# Class CreateChannel0UntriggerThresholdPayload

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

Represents an operator that creates a message payload that threshold value to release the lick detection state. Values above this threshold will untrigger a detected lick.

```
public class CreateChannel0UntriggerThresholdPayload
```

## Inheritance

[object](#) ← CreateChannel0UntriggerThresholdPayload

## Derived

[CreateTimestampedChannel0UntriggerThresholdPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Properties

### Channel0UntriggerThreshold

Gets or sets the value that threshold value to release the lick detection state. Values above this threshold will untrigger a detected lick.

```
public byte Channel0UntriggerThreshold { get; set; }
```

## Property Value

[byte](#)

## Methods

## GetMessage(MessageType)

Creates a message that threshold value to release the lick detection state. Values above this threshold will untrigger a detected lick.

```
public HarpMessage GetMessage(MessageType messageType)
```

### Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

### Returns

[HarpMessage](#)

A new message for the Channel0UntriggerThreshold register.

## GetPayload()

Creates a message payload for the Channel0UntriggerThreshold register.

```
public byte GetPayload()
```

### Returns

[byte](#)

The created message payload value.

# Class CreateLickStatePayload

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

Represents an operator that creates a message payload that emits an event when the state of any lick detector changes. Value will be High when lick detected and Low otherwise.

```
public class CreateLickStatePayload
```

## Inheritance

[object](#) ← CreateLickStatePayload

## Derived

[CreateTimestampedLickStatePayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## LickState

Gets or sets the value that emits an event when the state of any lick detector changes. Value will be High when lick detected and Low otherwise.

```
public LickChannels LickState { get; set; }
```

## Property Value

[LickChannels](#)

# Methods

## GetMessage(MessageType)

Creates a message that emits an event when the state of any lick detector changes. Value will be High when lick detected and Low otherwise.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the LickState register.

## GetPayload()

Creates a message payload for the LickState register.

```
public LickChannels GetPayload()
```

## Returns

[LickChannels](#)

The created message payload value.

# Class CreateMessage

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

Represents an operator which creates standard message payloads for the LicketySplit device.

```
public class CreateMessage : CreateMessageBuilder, IExpressionBuilder,  
ICustomTypeDescriptor, INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [CreateMessageBuilder](#) ← CreateMessage

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[CreateMessageBuilder.Build\(IEnumerable<Expression>\)](#), [CreateMessageBuilder.ArgumentRange](#),  
[CreateMessageBuilder.MessageType](#), [CreateMessageBuilder.Payload](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

## Constructors

### CreateMessage()

Initializes a new instance of the [CreateMessage](#) class.

```
public CreateMessage()
```

## See Also

[CreateLickStatePayload](#)

[CreateChannel0TriggerThresholdPayload](#)

[CreateChannel0UntriggerThresholdPayload](#)

# Class

# CreateTimestampedChannel0TriggerThresholdPayload

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

Represents an operator that creates a timestamped message payload that threshold value to detect the lick. Values below this threshold will be considered a detected lick.

```
public class CreateTimestampedChannel0TriggerThresholdPayload :  
CreateChannel0TriggerThresholdPayload
```

## Inheritance

[object](#) ← [CreateChannel0TriggerThresholdPayload](#) ←  
CreateTimestampedChannel0TriggerThresholdPayload

## Inherited Members

[CreateChannel0TriggerThresholdPayload.Channel0TriggerThreshold](#) ,  
[CreateChannel0TriggerThresholdPayload.GetPayload\(\)](#) ,  
[CreateChannel0TriggerThresholdPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that threshold value to detect the lick. Values below this threshold will be considered a detected lick.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

timestamp [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

Returns

[HarpMessage](#)

A new timestamped message for the Channel0TriggerThreshold register.

# Class

# CreateTimestampedChannel0UntriggerThresholdPayload

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

Represents an operator that creates a timestamped message payload that threshold value to release the lick detection state. Values above this threshold will untrigger a detected lick.

```
public class CreateTimestampedChannel0UntriggerThresholdPayload :  
CreateChannel0UntriggerThresholdPayload
```

## Inheritance

[object](#) ← [CreateChannel0UntriggerThresholdPayload](#) ←

CreateTimestampedChannel0UntriggerThresholdPayload

## Inherited Members

[CreateChannel0UntriggerThresholdPayload.Channel0UntriggerThreshold](#) ,  
[CreateChannel0UntriggerThresholdPayload.GetPayload\(\)](#) ,  
[CreateChannel0UntriggerThresholdPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that threshold value to release the lick detection state. Values above this threshold will untrigger a detected lick.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

timestamp [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

Returns

[HarpMessage](#)

A new timestamped message for the Channel0UntriggerThreshold register.

# Class CreateTimestampedLickStatePayload

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

Represents an operator that creates a timestamped message payload that emits an event when the state of any lick detector changes. Value will be High when lick detected and Low otherwise.

```
public class CreateTimestampedLickStatePayload : CreateLickStatePayload
```

## Inheritance

[object](#) ← [CreateLickStatePayload](#) ← CreateTimestampedLickStatePayload

## Inherited Members

[CreateLickStatePayload.LickState](#) , [CreateLickStatePayload.GetPayload\(\)](#) ,  
[CreateLickStatePayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that emits an event when the state of any lick detector changes. Value will be High when lick detected and Low otherwise.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#) ↗

A new timestamped message for the LickState register.

# Class Device

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

Represents an observable source of messages from the Harp device connected at the specified serial port.

```
[Combinator(MethodName = "Generate")]
[WorkflowElementCategory(ElementCategory.Source)]
public class Device : Device, INamedElement
```

## Inheritance

[object](#) ← [Source](#)<[HarpMessage](#)> ← [Device](#) ← [Device](#)

## Implements

[INamedElement](#)

## Inherited Members

[Device.Generate\(\)](#) , [Device.Generate\(IObservable<HarpMessage>\)](#) , [Device.OperationMode](#) ,  
[Device.OperationLed](#) , [Device.DumpRegisters](#) , [Device.VisualIndicators](#) , [Device.Heartbeat](#) ,  
[Device.IgnoreErrors](#) , [Device.PortName](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

# Constructors

## Device()

Initializes a new instance of the [Device](#) class.

```
public Device()
```

# Fields

## Metadata

Gets the contents of the metadata file describing the [AllenNeuralDynamics.LicketySplit](#) device registers.

```
public static readonly string Metadata
```

Field Value

[string](#)

## WhoAmI

Represents the unique identity class of the [AllenNeuralDynamics.LicketySplit](#) device. This field is constant.

```
public const int WhoAmI = 1400
```

Field Value

[int](#)

## Properties

### RegisterMap

Gets a read-only mapping from address to register type.

```
public static IReadOnlyDictionary<int, Type> RegisterMap { get; }
```

Property Value

[IReadOnlyDictionary](#)<[int](#), [Type](#)>

## Methods

### CreateAsync(string)

Initializes a new instance of the asynchronous API to configure and interface with LicketySplit devices on the specified serial port.

```
public static Task<AsyncDevice> CreateAsync(string portName)
```

## Parameters

**portName** [string](#)

The name of the serial port used to communicate with the Harp device.

## Returns

[Task](#) <[AsyncDevice](#)>

A task that represents the asynchronous initialization operation. The value of the [Result](#) parameter contains a new instance of the [AsyncDevice](#) class.

# Class FilterRegister

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

Represents an operator that filters register-specific messages reported by the [AllenNeuralDynamics.LicketySplit](#) device.

```
public class FilterRegister : FilterRegisterBuilder, IExpressionBuilder,  
ICustomTypeDescriptor, INamedElement
```

## Inheritance

```
object ↪ ← ExpressionBuilder ↪ ← HarpCombinatorBuilder ↪ ← RegisterCombinatorBuilder ↪ ←  
FilterRegisterBuilder ↪ ← FilterRegister
```

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

```
FilterRegisterBuilder.FilterType ↪ , FilterRegisterBuilder.Register ↪ ,  
RegisterCombinatorBuilder.Build(IEnumerable<Expression>) ↪ ,  
RegisterCombinatorBuilder.ArgumentRange ↪ , ExpressionBuilder.ToString() ↪ ,  
ExpressionBuilder.Unwrap(ExpressionBuilder) ↪ ,  
ExpressionBuilder.GetWorkflowElement(ExpressionBuilder) ↪ ,  
ExpressionBuilder.GetVisualizerElement(ExpressionBuilder) ↪ ,  
ExpressionBuilder.GetVisualizerMappings(ExpressionBuilder) ↪ ,  
ExpressionBuilder.FromWorkflowElement(object, ElementCategory) ↪ ,  
ExpressionBuilder.GetElementDisplayName(Type) ↪ ,  
ExpressionBuilder.GetElementDisplayName(object) ↪ , object.Equals(object) ↪ ,  
object.Equals(object, object) ↪ , object.ReferenceEquals(object, object) ↪ , object.GetHashCode() ↪ ,  
object.GetType() ↪ , object.MemberwiseClone() ↪
```

## Constructors

### FilterRegister()

Initializes a new instance of the [FilterRegister](#) class.

```
public FilterRegister()
```

## See Also

[LickState](#)

[Channel0TriggerThreshold](#)

[Channel0UntriggerThreshold](#)

# Class Format

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

Represents an operator which formats a sequence of values as specific LicketySplit register messages.

```
public class Format : FormatBuilder, IExpressionBuilder, ICustomTypeDescriptor,  
INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [FormatBuilder](#) ← Format

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[FormatBuilder.Build\(IEnumerable<Expression>\)](#), [FormatBuilder.ArgumentRange](#),  
[FormatBuilder.MessageType](#), [FormatBuilder.Register](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

# Constructors

## Format()

Initializes a new instance of the [Format](#) class.

```
public Format()
```

## See Also

[LickState](#)

[Channel0TriggerThreshold](#)

[Channel0UntriggerThreshold](#)

# Class GetMetadata

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

Represents an operator that returns the contents of the metadata file describing the [AllenNeuralDynamics.LicketySplit](#) device registers.

```
public class GetMetadata : Source<string>
```

## Inheritance

[object](#) ← [Source](#)<[string](#)> ← GetMetadata

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Methods

### Generate()

Returns an observable sequence with the contents of the metadata file describing the [AllenNeuralDynamics.LicketySplit](#) device registers.

```
public override IObservable<string> Generate()
```

## Returns

[IObservable](#)<[string](#)>

A sequence with a single [string](#) object representing the contents of the metadata file.

# Class GroupByRegister

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

Represents an operator that groups the sequence of [AllenNeuralDynamics.LicketySplit](#)" messages by register type.

```
public class GroupByRegister : Combinator<HarpMessage, IGroupedObservable<Type, HarpMessage>>
```

## Inheritance

```
object ↗ ← Combinator ↗ <HarpMessage ↗, IGroupedObservable ↗ <Type ↗, HarpMessage ↗>> ← GroupByRegister
```

## Inherited Members

```
Combinator<HarpMessage, IGroupedObservable<Type, HarpMessage>>.Process(I Observable<HarpMessage>) ↗ ,  
object.ToString() ↗ , object.Equals(object) ↗ , object.Equals(object, object) ↗ ,  
object.ReferenceEquals(object, object) ↗ , object.GetHashCode() ↗ , object.GetType() ↗ ,  
object.MemberwiseClone() ↗
```

## Methods

### Process(I Observable<HarpMessage>)

Groups an observable sequence of [AllenNeuralDynamics.LicketySplit](#) messages by register type.

```
public override IObservable<IGroupedObservable<Type, HarpMessage>>  
Process(IObservable<HarpMessage> source)
```

## Parameters

source [IObservable](#)<HarpMessage>

The sequence of Harp device messages.

## Returns

[IObservable](#) < [IGroupedObservable](#) < [Type](#), [HarpMessage](#) >>

A sequence of observable groups, each of which corresponds to a unique [AllenNeuralDynamics.Lickety.Split](#) register.

# Enum LickChannels

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

The channel of the lick detector.

```
[Flags]
public enum LickChannels : byte
```

## Fields

Channel0 = 1

Channel1 = 2

None = 0

# Class LickState

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

Represents a register that emits an event when the state of any lick detector changes. Value will be High when lick detected and Low otherwise.

```
public class LickState
```

## Inheritance

[object](#) ← LickState

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [LickState](#) register. This field is constant.

```
public const int Address = 32
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [LickState](#) register. This field is constant.

```
public const int RegisterLength = 1
```

## Field Value

[int↗](#)

## RegisterType

Represents the payload type of the [LickState](#) register. This field is constant.

```
public const PayloadType RegisterType = U8
```

## Field Value

[PayloadType↗](#)

## Methods

### FromPayload(MessageType, LickChannels)

Returns a Harp message for the [LickState](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, LickChannels value)
```

#### Parameters

**messageType** [MessageType↗](#)

The type of the Harp message.

**value** [LickChannels](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage↗](#)

A [HarpMessage↗](#) object for the [LickState](#) register with the specified message type and payload.

## FromPayload(double, MessageType, LickChannels)

Returns a timestamped Harp message for the [LickState](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
LickChannels value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [LickChannels](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [LickState](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [LickState](#) register messages.

```
public static LickChannels GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## [LickChannels](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [LickState](#) register messages.

```
public static Timestamped<LickChannels> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <[LickChannels](#)>

A value representing the timestamped message payload.

# Class Parse

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

Represents an operator which filters and selects specific messages reported by the LicketySplit device.

```
public class Parse : ParseBuilder, IExpressionBuilder, ICustomTypeDescriptor, INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [RegisterCombinatorBuilder](#) ← [ParseBuilder](#) ← Parse

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[ParseBuilder.Register](#), [RegisterCombinatorBuilder.Build\(IEnumerable<Expression>\)](#),  
[RegisterCombinatorBuilder.ArgumentRange](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

## Constructors

### Parse()

Initializes a new instance of the [Parse](#) class.

```
public Parse()
```

## See Also

[LickState](#)

[Channel0TriggerThreshold](#)

[Channel0UntriggerThreshold](#)

# Class TimestampedChannel0TriggerThreshold

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

Provides methods for manipulating timestamped messages from the Channel0TriggerThreshold register.

```
public class TimestampedChannel0TriggerThreshold
```

## Inheritance

[object](#) ← TimestampedChannel0TriggerThreshold

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [Channel0TriggerThreshold](#) register. This field is constant.

```
public const int Address = 33
```

### Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [Channel0TriggerThreshold](#) register messages.

```
public static Timestamped<byte> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#)<[byte](#)>

A value representing the timestamped message payload.

## See Also

[Channel0TriggerThreshold](#)

# Class

# TimestampedChannel0UntriggerThreshold

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

Provides methods for manipulating timestamped messages from the Channel0UntriggerThreshold register.

```
public class TimestampedChannel0UntriggerThreshold
```

## Inheritance

[object](#) ← TimestampedChannel0UntriggerThreshold

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [Channel0UntriggerThreshold](#) register. This field is constant.

```
public const int Address = 34
```

### Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [Channel0UntriggerThreshold](#) register messages.

```
public static Timestamped<byte> GetPayload(HarpMessage message)
```

## Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#)<[byte](#)>

A value representing the timestamped message payload.

## See Also

[Channel0UntriggerThreshold](#)

# Class TimestampedLickState

Namespace: [AllenNeuralDynamics.LicketySplit](#)

Assembly: AllenNeuralDynamics.LicketySplit.dll

Provides methods for manipulating timestamped messages from the LickState register.

```
public class TimestampedLickState
```

## Inheritance

[object](#) ← TimestampedLickState

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [LickState](#) register. This field is constant.

```
public const int Address = 32
```

## Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [LickState](#) register messages.

```
public static Timestamped<LickChannels> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[LickChannels](#)>

A value representing the timestamped message payload.

## See Also

[LickState](#)

# Namespace AllenNeuralDynamics.SniffDetector

## Classes

### [AsyncDevice](#)

Represents an asynchronous API to configure and interface with SniffDetector devices.

### [CreateMessage](#)

Represents an operator which creates standard message payloads for the SniffDetector device.

### [CreateRawVoltageDispatchRatePayload](#)

Represents an operator that creates a message payload that sets the rate at which the RawVoltage event is emitted.

### [CreateRawVoltagePayload](#)

Represents an operator that creates a message payload that emits a periodic event containing the raw voltage read of the thermistor sensor.

### [CreateTimestampedRawVoltageDispatchRatePayload](#)

Represents an operator that creates a timestamped message payload that sets the rate at which the RawVoltage event is emitted.

### [CreateTimestampedRawVoltagePayload](#)

Represents an operator that creates a timestamped message payload that emits a periodic event containing the raw voltage read of the thermistor sensor.

### [Device](#)

Represents an observable source of messages from the Harp device connected at the specified serial port.

### [FilterRegister](#)

Represents an operator that filters register-specific messages reported by the [AllenNeuralDynamics.SniffDetector](#) device.

### [Format](#)

Represents an operator which formats a sequence of values as specific SniffDetector register messages.

### [GetMetadata](#)

Represents an operator that returns the contents of the metadata file describing the [AllenNeuralDynamics.SniffDetector](#) device registers.

### [GroupByRegister](#)

Represents an operator that groups the sequence of [AllenNeuralDynamics.SniffDetector](#)" messages by register type.

#### [Parse](#)

Represents an operator which filters and selects specific messages reported by the SniffDetector device.

#### [RawVoltage](#)

Represents a register that emits a periodic event containing the raw voltage read of the thermistor sensor.

#### [RawVoltageDispatchRate](#)

Represents a register that sets the rate at which the RawVoltage event is emitted.

#### [TimestampedRawVoltage](#)

Provides methods for manipulating timestamped messages from the RawVoltage register.

#### [TimestampedRawVoltageDispatchRate](#)

Provides methods for manipulating timestamped messages from the RawVoltageDispatchRate register.

# Class AsyncDevice

Namespace: [AllenNeuralDynamics.SniffDetector](#)

Assembly: AllenNeuralDynamics.SniffDetector.dll

Represents an asynchronous API to configure and interface with SniffDetector devices.

```
public class AsyncDevice : AsyncDevice, IDisposable
```

## Inheritance

[object](#) ← [AsyncDevice](#) ← AsyncDevice

## Implements

[IDisposable](#)

## Inherited Members

[AsyncDevice.ReadWhoAmIAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadHardwareVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadAssemblyVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadCoreVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadFirmwareVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadTimestampSecondsAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadTimestampMicrosecondsAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadOperationControlAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadResetDeviceAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadDeviceNameAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadSerialNumberAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadClockConfigurationAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadByteAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadByteArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSByteAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSByteArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt16Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt16ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt16Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt16ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt32Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt32ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt32Async\(int, CancellationToken\)](#) ,

[AsyncDevice.ReadInt32ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt64Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt64ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt64Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt64ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSingleAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSingleArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.WriteTimestampSecondsAsync\(uint, CancellationToken\)](#) ,  
[AsyncDevice.WriteResetDeviceAsync\(ResetFlags, CancellationToken\)](#) ,  
[AsyncDevice.WriteDeviceNameAsync\(string, CancellationToken\)](#) ,  
[AsyncDevice.WriteClockConfigurationAsync\(ClockConfigurationFlags, CancellationToken\)](#) ,  
[AsyncDevice.WriteByteAsync\(int, byte, CancellationToken\)](#) ,  
[AsyncDevice.WriteByteAsync\(int, byte\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteSByteAsync\(int, sbyte, CancellationToken\)](#) ,  
[AsyncDevice.WriteSByteAsync\(int, sbyte\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteUInt16Async\(int, ushort, CancellationToken\)](#) ,  
[AsyncDevice.WriteUInt16Async\(int, ushort\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine16Async\(int, short, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine16Async\(int, short\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, uint, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, uint\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, int, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, int\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, ulong, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, ulong\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, long, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, long\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteSingleAsync\(int, float, CancellationToken\)](#) ,  
[AsyncDevice.WriteSingleAsync\(int, float\[\], CancellationToken\)](#) ,  
[AsyncDevice.CommandAsync\(HarpMessage, CancellationToken\)](#) , [AsyncDevice.Dispose\(\)](#) ,  
[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Methods

### ReadRawVoltageAsync(CancellationToken)

Asynchronously reads the contents of the RawVoltage register.

```
public Task<ushort> ReadRawVoltageAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken  [CancellationToken](#)

A  [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<ushort>

A task that represents the asynchronous read operation. The  [Result](#) property contains the register payload.

## ReadRawVoltageDispatchRateAsync(CancellationToken)

Asynchronously reads the contents of the RawVoltageDispatchRate register.

```
public Task<ushort> ReadRawVoltageDispatchRateAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken  [CancellationToken](#)

A  [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<ushort>

A task that represents the asynchronous read operation. The  [Result](#) property contains the register payload.

## ReadTimestampedRawVoltageAsync(CancellationToken)

Asynchronously reads the timestamped contents of the RawVoltage register.

```
public Task<Timestamped<ushort>> ReadTimestampedRawVoltageAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<ushort>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedRawVoltageDispatchRateAsync(CancellationTokenT oken)

Asynchronously reads the timestamped contents of the RawVoltageDispatchRate register.

```
public Task<Timestamped<ushort>>  
ReadTimestampedRawVoltageDispatchRateAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<ushort>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## WriteRawVoltageDispatchRateAsync(ushort, CancellationToken)

Asynchronously writes a value to the RawVoltageDispatchRate register.

```
public Task WriteRawVoltageDispatchRateAsync(ushort value, CancellationToken  
cancellationToken = default)
```

## Parameters

**value** [ushort](#)

The value to be stored in the register.

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)

The task object representing the asynchronous write operation.

# Class CreateMessage

Namespace: [AllenNeuralDynamics.SniffDetector](#)

Assembly: AllenNeuralDynamics.SniffDetector.dll

Represents an operator which creates standard message payloads for the SniffDetector device.

```
public class CreateMessage : CreateMessageBuilder, IExpressionBuilder,  
ICustomTypeDescriptor, INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [CreateMessageBuilder](#) ← CreateMessage

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[CreateMessageBuilder.Build\(IEnumerable<Expression>\)](#), [CreateMessageBuilder.ArgumentRange](#),  
[CreateMessageBuilder.MessageType](#), [CreateMessageBuilder.Payload](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

## Constructors

### CreateMessage()

Initializes a new instance of the [CreateMessage](#) class.

```
public CreateMessage()
```

## See Also

[CreateRawVoltagePayload](#)

[CreateRawVoltageDispatchRatePayload](#)

# Class CreateRawVoltageDispatchRatePayload

Namespace: [AllenNeuralDynamics.SniffDetector](#)

Assembly: AllenNeuralDynamics.SniffDetector.dll

Represents an operator that creates a message payload that sets the rate at which the RawVoltage event is emitted.

```
public class CreateRawVoltageDispatchRatePayload
```

## Inheritance

[object](#) ← CreateRawVoltageDispatchRatePayload

## Derived

[CreateTimestampedRawVoltageDispatchRatePayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Properties

### RawVoltageDispatchRate

Gets or sets the value that sets the rate at which the RawVoltage event is emitted.

```
public ushort RawVoltageDispatchRate { get; set; }
```

## Property Value

[ushort](#)

## Methods

### GetMessage(MessageType)

Creates a message that sets the rate at which the RawVoltage event is emitted.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the RawVoltageDispatchRate register.

## GetPayload()

Creates a message payload for the RawVoltageDispatchRate register.

```
public ushort GetPayload()
```

## Returns

[ushort](#)

The created message payload value.

# Class CreateRawVoltagePayload

Namespace: [AllenNeuralDynamics.SniffDetector](#)

Assembly: AllenNeuralDynamics.SniffDetector.dll

Represents an operator that creates a message payload that emits a periodic event containing the raw voltage read of the thermistor sensor.

```
public class CreateRawVoltagePayload
```

## Inheritance

[object](#) ← CreateRawVoltagePayload

## Derived

[CreateTimestampedRawVoltagePayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## RawVoltage

Gets or sets the value that emits a periodic event containing the raw voltage read of the thermistor sensor.

```
public ushort RawVoltage { get; set; }
```

## Property Value

[ushort](#)

# Methods

## GetMessage(MessageType)

Creates a message that emits a periodic event containing the raw voltage read of the thermistor sensor.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the RawVoltage register.

## GetPayload()

Creates a message payload for the RawVoltage register.

```
public ushort GetPayload()
```

## Returns

[ushort](#)

The created message payload value.

# Class

# CreateTimestampedRawVoltageDispatchRatePayload

Namespace: [AllenNeuralDynamics.SniffDetector](#)

Assembly: AllenNeuralDynamics.SniffDetector.dll

Represents an operator that creates a timestamped message payload that sets the rate at which the RawVoltage event is emitted.

```
public class CreateTimestampedRawVoltageDispatchRatePayload :  
CreateRawVoltageDispatchRatePayload
```

## Inheritance

[object](#) ← [CreateRawVoltageDispatchRatePayload](#) ←  
CreateTimestampedRawVoltageDispatchRatePayload

## Inherited Members

[CreateRawVoltageDispatchRatePayload.RawVoltageDispatchRate](#) ,  
[CreateRawVoltageDispatchRatePayload.GetPayload\(\)](#) ,  
[CreateRawVoltageDispatchRatePayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that sets the rate at which the RawVoltage event is emitted.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

Returns

[HarpMessage](#)

A new timestamped message for the RawVoltageDispatchRate register.

# Class CreateTimestampedRawVoltagePayload

Namespace: [AllenNeuralDynamics.SniffDetector](#)

Assembly: AllenNeuralDynamics.SniffDetector.dll

Represents an operator that creates a timestamped message payload that emits a periodic event containing the raw voltage read of the thermistor sensor.

```
public class CreateTimestampedRawVoltagePayload : CreateRawVoltagePayload
```

## Inheritance

[object](#) ← [CreateRawVoltagePayload](#) ← CreateTimestampedRawVoltagePayload

## Inherited Members

[CreateRawVoltagePayload.RawVoltage](#) , [CreateRawVoltagePayload.GetPayload\(\)](#) ,  
[CreateRawVoltagePayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that emits a periodic event containing the raw voltage read of the thermistor sensor.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#) ↗

A new timestamped message for the RawVoltage register.

# Class Device

Namespace: [AllenNeuralDynamics.SniffDetector](#)

Assembly: AllenNeuralDynamics.SniffDetector.dll

Represents an observable source of messages from the Harp device connected at the specified serial port.

```
[Combinator(MethodName = "Generate")]
[WorkflowElementCategory(ElementCategory.Source)]
public class Device : Device, INamedElement
```

## Inheritance

[object](#) ← [Source](#)<[HarpMessage](#)> ← [Device](#) ← [Device](#)

## Implements

[INamedElement](#)

## Inherited Members

[Device.Generate\(\)](#) , [Device.Generate\(IObservable<HarpMessage>\)](#) , [Device.OperationMode](#) ,  
[Device.OperationLed](#) , [Device.DumpRegisters](#) , [Device.VisualIndicators](#) , [Device.Heartbeat](#) ,  
[Device.IgnoreErrors](#) , [Device.PortName](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

# Constructors

## Device()

Initializes a new instance of the [Device](#) class.

```
public Device()
```

# Fields

## Metadata

Gets the contents of the metadata file describing the [AllenNeuralDynamics.SniffDetector](#) device registers.

```
public static readonly string Metadata
```

Field Value

[string](#)

## WhoAmI

Represents the unique identity class of the [AllenNeuralDynamics.SniffDetector](#) device. This field is constant.

```
public const int WhoAmI = 1401
```

Field Value

[int](#)

## Properties

### RegisterMap

Gets a read-only mapping from address to register type.

```
public static IReadOnlyDictionary<int, Type> RegisterMap { get; }
```

Property Value

[IReadOnlyDictionary](#)<[int](#), [Type](#)>

## Methods

### CreateAsync(string)

Initializes a new instance of the asynchronous API to configure and interface with SniffDetector devices on the specified serial port.

```
public static Task<AsyncDevice> CreateAsync(string portName)
```

## Parameters

**portName** [string](#)

The name of the serial port used to communicate with the Harp device.

## Returns

[Task](#) <[AsyncDevice](#)>

A task that represents the asynchronous initialization operation. The value of the [Result](#) parameter contains a new instance of the [AsyncDevice](#) class.

# Class FilterRegister

Namespace: [AllenNeuralDynamics.SniffDetector](#)

Assembly: AllenNeuralDynamics.SniffDetector.dll

Represents an operator that filters register-specific messages reported by the [AllenNeuralDynamics.SniffDetector](#) device.

```
public class FilterRegister : FilterRegisterBuilder, IExpressionBuilder,  
ICustomTypeDescriptor, INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [RegisterCombinatorBuilder](#) ←  
[FilterRegisterBuilder](#) ← FilterRegister

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[FilterRegisterBuilder.FilterType](#), [FilterRegisterBuilder.Register](#),  
[RegisterCombinatorBuilder.Build\(IEnumerable<Expression>\)](#),  
[RegisterCombinatorBuilder.ArgumentRange](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

## Constructors

### FilterRegister()

Initializes a new instance of the [FilterRegister](#) class.

```
public FilterRegister()
```

## See Also

[RawVoltage](#)

[RawVoltageDispatchRate](#)

# Class Format

Namespace: [AllenNeuralDynamics.SniffDetector](#)

Assembly: AllenNeuralDynamics.SniffDetector.dll

Represents an operator which formats a sequence of values as specific SniffDetector register messages.

```
public class Format : FormatBuilder, IExpressionBuilder, ICustomTypeDescriptor,  
INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [FormatBuilder](#) ← Format

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[FormatBuilder.Build\(IEnumerable<Expression>\)](#), [FormatBuilder.ArgumentRange](#),  
[FormatBuilder.MessageType](#), [FormatBuilder.Register](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

# Constructors

## Format()

Initializes a new instance of the [Format](#) class.

```
public Format()
```

## See Also

[RawVoltage](#)

[RawVoltageDispatchRate](#)

# Class GetMetadata

Namespace: [AllenNeuralDynamics.SniffDetector](#)

Assembly: AllenNeuralDynamics.SniffDetector.dll

Represents an operator that returns the contents of the metadata file describing the [AllenNeuralDynamics.SniffDetector](#) device registers.

```
public class GetMetadata : Source<string>
```

## Inheritance

[object](#) ← [Source](#)<[string](#)> ← GetMetadata

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Methods

### Generate()

Returns an observable sequence with the contents of the metadata file describing the [AllenNeuralDynamics.SniffDetector](#) device registers.

```
public override IObservable<string> Generate()
```

## Returns

[IObservable](#)<[string](#)>

A sequence with a single [string](#) object representing the contents of the metadata file.

# Class GroupByRegister

Namespace: [AllenNeuralDynamics.SniffDetector](#)

Assembly: AllenNeuralDynamics.SniffDetector.dll

Represents an operator that groups the sequence of [AllenNeuralDynamics.SniffDetector](#)" messages by register type.

```
public class GroupByRegister : Combinator<HarpMessage, IGroupedObservable<Type, HarpMessage>>
```

## Inheritance

```
object ↗ ← Combinator ↗ <HarpMessage ↗, IGroupedObservable ↗ <Type ↗, HarpMessage ↗>> ← GroupByRegister
```

## Inherited Members

```
Combinator<HarpMessage, IGroupedObservable<Type, HarpMessage>>.Process(I Observable<HarpMessage>) ↗ ,  
object.ToString() ↗ , object.Equals(object) ↗ , object.Equals(object, object) ↗ ,  
object.ReferenceEquals(object, object) ↗ , object.GetHashCode() ↗ , object.GetType() ↗ ,  
object.MemberwiseClone() ↗
```

## Methods

### Process(I Observable<HarpMessage>)

Groups an observable sequence of [AllenNeuralDynamics.SniffDetector](#) messages by register type.

```
public override IObservable<IGroupedObservable<Type, HarpMessage>>  
Process(IObservable<HarpMessage> source)
```

## Parameters

source [IObservable](#)<HarpMessage>

The sequence of Harp device messages.

## Returns

[IObservable](#) < [IGroupedObservable](#) < [Type](#), [HarpMessage](#) >>

A sequence of observable groups, each of which corresponds to a unique [AllenNeuralDynamics.SniffDetector](#) register.

# Class Parse

Namespace: [AllenNeuralDynamics.SniffDetector](#)

Assembly: AllenNeuralDynamics.SniffDetector.dll

Represents an operator which filters and selects specific messages reported by the SniffDetector device.

```
public class Parse : ParseBuilder, IExpressionBuilder, ICustomTypeDescriptor, INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [RegisterCombinatorBuilder](#) ← [ParseBuilder](#) ← Parse

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[ParseBuilder.Register](#), [RegisterCombinatorBuilder.Build\(IEnumerable<Expression>\)](#),  
[RegisterCombinatorBuilder.ArgumentRange](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

# Constructors

## Parse()

Initializes a new instance of the [Parse](#) class.

```
public Parse()
```

## See Also

[RawVoltage](#)

[RawVoltageDispatchRate](#)

# Class RawVoltage

Namespace: [AllenNeuralDynamics.SniffDetector](#)

Assembly: AllenNeuralDynamics.SniffDetector.dll

Represents a register that emits a periodic event containing the raw voltage read of the thermistor sensor.

```
public class RawVoltage
```

## Inheritance

[object](#) ← RawVoltage

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [RawVoltage](#) register. This field is constant.

```
public const int Address = 32
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [RawVoltage](#) register. This field is constant.

```
public const int RegisterLength = 1
```

## Field Value

[int↗](#)

## RegisterType

Represents the payload type of the [RawVoltage](#) register. This field is constant.

```
public const PayloadType RegisterType = U16
```

## Field Value

[PayloadType↗](#)

## Methods

### FromPayload(MessageType, ushort)

Returns a Harp message for the [RawVoltage](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, ushort value)
```

#### Parameters

**messageType** [MessageType↗](#)

The type of the Harp message.

**value** [ushort↗](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage↗](#)

A [HarpMessage↗](#) object for the [RawVoltage](#) register with the specified message type and payload.

## FromPayload(double, MessageType, ushort)

Returns a timestamped Harp message for the [RawVoltage](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
ushort value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [ushort](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [RawVoltage](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [RawVoltage](#) register messages.

```
public static ushort GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## [ushort](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [RawVoltage](#) register messages.

```
public static Timestamped<ushort> GetTimestampedPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) < ushort >

A value representing the timestamped message payload.

# Class RawVoltageDispatchRate

Namespace: [AllenNeuralDynamics.SniffDetector](#)

Assembly: AllenNeuralDynamics.SniffDetector.dll

Represents a register that sets the rate at which the RawVoltage event is emitted.

```
public class RawVoltageDispatchRate
```

## Inheritance

[object](#) ← RawVoltageDispatchRate

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [RawVoltageDispatchRate](#) register. This field is constant.

```
public const int Address = 33
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [RawVoltageDispatchRate](#) register. This field is constant.

```
public const int RegisterLength = 1
```

### Field Value

## .RegisterType

Represents the payload type of the [RawVoltageDispatchRate](#) register. This field is constant.

```
public const PayloadType RegisterType = U16
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, ushort)

Returns a Harp message for the [RawVoltageDispatchRate](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, ushort value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [ushort](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [RawVoltageDispatchRate](#) register with the specified message type and payload.

## FromPayload(double, MessageType, ushort)

Returns a timestamped Harp message for the [RawVoltageDispatchRate](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
ushort value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [ushort](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [RawVoltageDispatchRate](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [RawVoltageDispatchRate](#) register messages.

```
public static ushort GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## [ushort](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [RawVoltageDispatchRate](#) register messages.

```
public static Timestamped<ushort> GetTimestampedPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) < ushort >

A value representing the timestamped message payload.

# Class TimestampedRawVoltage

Namespace: [AllenNeuralDynamics.SniffDetector](#)

Assembly: AllenNeuralDynamics.SniffDetector.dll

Provides methods for manipulating timestamped messages from the RawVoltage register.

```
public class TimestampedRawVoltage
```

## Inheritance

[object](#) ← TimestampedRawVoltage

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [RawVoltage](#) register. This field is constant.

```
public const int Address = 32
```

## Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [RawVoltage](#) register messages.

```
public static Timestamped<ushort> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[ushort](#)>

A value representing the timestamped message payload.

## See Also

[RawVoltage](#)

# Class TimestampedRawVoltageDispatchRate

Namespace: [AllenNeuralDynamics.SniffDetector](#)

Assembly: AllenNeuralDynamics.SniffDetector.dll

Provides methods for manipulating timestamped messages from the RawVoltageDispatchRate register.

```
public class TimestampedRawVoltageDispatchRate
```

## Inheritance

[object](#) ← TimestampedRawVoltageDispatchRate

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [RawVoltageDispatchRate](#) register. This field is constant.

```
public const int Address = 33
```

### Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [RawVoltageDispatchRate](#) register messages.

```
public static Timestamped<ushort> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[ushort](#)>

A value representing the timestamped message payload.

## See Also

[RawVoltageDispatchRate](#)

# Namespace AllenNeuralDynamics.Treadmill

## Classes

### [AsyncDevice](#)

Represents an asynchronous API to configure and interface with Treadmill devices.

### [BrakeCurrentSetPoint](#)

Represents a register that sets the raw value of the torque set-point to be applied to the treadmill. This value is cleared to 0 if torque\_limiting is enabled and triggered. Further writes in this condition return a WRITE\_ERROR.

### [CreateBrakeCurrentSetPointPayload](#)

Represents an operator that creates a message payload that sets the raw value of the torque set-point to be applied to the treadmill. This value is cleared to 0 if torque\_limiting is enabled and triggered. Further writes in this condition return a WRITE\_ERROR.

### [CreateEnableTorqueLimitPayload](#)

Represents an operator that creates a message payload that enables(1)/Disables(0) the brake if the maximum torque sensor value is detected. This register will be enabled by default.

### [CreateEncoderPayload](#)

Represents an operator that creates a message payload that contains the current accumulated number of ticks.

### [CreateMessage](#)

Represents an operator which creates standard message payloads for the Treadmill device.

### [CreateResetTareSensorsPayload](#)

Represents an operator that creates a message payload that removes the tare from the specified sensors.

### [CreateSensorDataDispatchRatePayload](#)

Represents an operator that creates a message payload that value greater than 0 will enable the periodic dispatch of treadmill data events at the specified rate (sp/s).

### [CreateSensorDataPayload](#)

Represents an operator that creates a message payload that emits a periodic event containing the packaged treadmill data. [Encoder, Torque, TorqueLoadCurrent].

### [CreateTareSensorsPayload](#)

Represents an operator that creates a message payload that tares the specified sensors.

### [CreateTimestampedBrakeCurrentSetPointPayload](#)

Represents an operator that creates a timestamped message payload that sets the raw value of the torque set-point to be applied to the treadmill. This value is cleared to 0 if torque\_limiting is enabled and triggered. Further writes in this condition return a WRITE\_ERROR.

#### [CreateTimestampedEnableTorqueLimitPayload](#)

Represents an operator that creates a timestamped message payload that enables(1)/Disables(0) the brake if the maximum torque sensor value is detected. This register will be enabled by default.

#### [CreateTimestampedEncoderPayload](#)

Represents an operator that creates a timestamped message payload that contains the current accumulated number of ticks.

#### [CreateTimestampedResetTareSensorsPayload](#)

Represents an operator that creates a timestamped message payload that removes the tare from the specified sensors.

#### [CreateTimestampedSensorDataDispatchRatePayload](#)

Represents an operator that creates a timestamped message payload that value greater than 0 will enable the periodic dispatch of treadmill data events at the specified rate (sp/s).

#### [CreateTimestampedSensorDataPayload](#)

Represents an operator that creates a timestamped message payload that emits a periodic event containing the packaged treadmill data. [Encoder, Torque, TorqueLoadCurrent].

#### [CreateTimestampedTareSensorsPayload](#)

Represents an operator that creates a timestamped message payload that tares the specified sensors.

#### [CreateTimestampedTorqueLimitStatePayload](#)

Represents an operator that creates a timestamped message payload that a value greater than 1 indicates that the torque limit has been triggered and the brake setpoint will be cleared. Writing a value of 0 will clear the torque limit state and re-enable the brake.

#### [CreateTimestampedTorqueLoadCurrentPayload](#)

Represents an operator that creates a timestamped message payload that contains the current output current applied to the variable torque load.

#### [CreateTimestampedTorquePayload](#)

Represents an operator that creates a timestamped message payload that contains the current torque value.

#### [CreateTorqueLimitStatePayload](#)

Represents an operator that creates a message payload that a value greater than 1 indicates that the torque limit has been triggered and the brake setpoint will be cleared. Writing a value of 0 will clear the torque limit state and re-enable the brake.

## [CreateTorqueLoadCurrentPayload](#)

Represents an operator that creates a message payload that contains the current output current applied to the variable torque load.

## [CreateTorquePayload](#)

Represents an operator that creates a message payload that contains the current torque value.

## [Device](#)

Represents an observable source of messages from the Harp device connected at the specified serial port.

## [EnableTorqueLimit](#)

Represents a register that enables(1)/Disables(0) the brake if the maximum torque sensor value is detected. This register will be enabled by default.

## [Encoder](#)

Represents a register that contains the current accumulated number of ticks.

## [FilterRegister](#)

Represents an operator that filters register-specific messages reported by the [AllenNeuralDynamics.Treadmill](#) device.

## [Format](#)

Represents an operator which formats a sequence of values as specific Treadmill register messages.

## [GetMetadata](#)

Represents an operator that returns the contents of the metadata file describing the [AllenNeuralDynamics.Treadmill](#) device registers.

## [GroupByRegister](#)

Represents an operator that groups the sequence of [AllenNeuralDynamics.Treadmill](#) messages by register type.

## [Parse](#)

Represents an operator which filters and selects specific messages reported by the Treadmill device.

## [ResetTareSensors](#)

Represents a register that removes the tare from the specified sensors.

## [SensorData](#)

Represents a register that emits a periodic event containing the packaged treadmill data. [Encoder, Torque, TorqueLoadCurrent].

## [SensorDataDispatchRate](#)

Represents a register that value greater than 0 will enable the periodic dispatch of treadmill data events at the specified rate (sp/s).

### [TareSensors](#)

Represents a register that tares the specified sensors.

### [TimestampedBrakeCurrentSetPoint](#)

Provides methods for manipulating timestamped messages from the BrakeCurrentSetPoint register.

### [TimestampedEnableTorqueLimit](#)

Provides methods for manipulating timestamped messages from the EnableTorqueLimit register.

### [TimestampedEncoder](#)

Provides methods for manipulating timestamped messages from the Encoder register.

### [TimestampedResetTareSensors](#)

Provides methods for manipulating timestamped messages from the ResetTareSensors register.

### [TimestampedSensorData](#)

Provides methods for manipulating timestamped messages from the SensorData register.

### [TimestampedSensorDataDispatchRate](#)

Provides methods for manipulating timestamped messages from the SensorDataDispatchRate register.

### [TimestampedTareSensors](#)

Provides methods for manipulating timestamped messages from the TareSensors register.

### [TimestampedTorque](#)

Provides methods for manipulating timestamped messages from the Torque register.

### [TimestampedTorqueLimitState](#)

Provides methods for manipulating timestamped messages from the TorqueLimitState register.

### [TimestampedTorqueLoadCurrent](#)

Provides methods for manipulating timestamped messages from the TorqueLoadCurrent register.

### [Torque](#)

Represents a register that contains the current torque value.

### [TorqueLimitState](#)

Represents a register that a value greater than 1 indicates that the torque limit has been triggered and the brake setpoint will be cleared. Writing a value of 0 will clear the torque limit state and re-enable the brake.

### [TorqueLoadCurrent](#)

Represents a register that contains the current output current applied to the variable torque load.

## Structs

### [SensorDataPayload](#)

Represents the payload of the SensorData register.

## Enums

### [Sensors](#)

Available sensors.

# Class AsyncDevice

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an asynchronous API to configure and interface with Treadmill devices.

```
public class AsyncDevice : AsyncDevice, IDisposable
```

## Inheritance

[object](#) ← [AsyncDevice](#) ← AsyncDevice

## Implements

[IDisposable](#)

## Inherited Members

[AsyncDevice.ReadWhoAmIAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadHardwareVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadAssemblyVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadCoreVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadFirmwareVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadTimestampSecondsAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadTimestampMicrosecondsAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadOperationControlAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadResetDeviceAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadDeviceNameAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadSerialNumberAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadClockConfigurationAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadByteAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadByteArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSByteAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSByteArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt16Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt16ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt16Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt16ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt32Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt32ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt32Async\(int, CancellationToken\)](#) ,

[AsyncDevice.ReadInt32ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt64Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt64ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt64Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt64ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSingleAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSingleArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.WriteTimestampSecondsAsync\(uint, CancellationToken\)](#) ,  
[AsyncDevice.WriteResetDeviceAsync\(ResetFlags, CancellationToken\)](#) ,  
[AsyncDevice.WriteDeviceNameAsync\(string, CancellationToken\)](#) ,  
[AsyncDevice.WriteClockConfigurationAsync\(ClockConfigurationFlags, CancellationToken\)](#) ,  
[AsyncDevice.WriteByteAsync\(int, byte, CancellationToken\)](#) ,  
[AsyncDevice.WriteByteAsync\(int, byte\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteSByteAsync\(int, sbyte, CancellationToken\)](#) ,  
[AsyncDevice.WriteSByteAsync\(int, sbyte\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteUInt16Async\(int, ushort, CancellationToken\)](#) ,  
[AsyncDevice.WriteUInt16Async\(int, ushort\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine16Async\(int, short, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine16Async\(int, short\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, uint, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, uint\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, int, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, int\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, ulong, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, ulong\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, long, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, long\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteSingleAsync\(int, float, CancellationToken\)](#) ,  
[AsyncDevice.WriteSingleAsync\(int, float\[\], CancellationToken\)](#) ,  
[AsyncDevice.CommandAsync\(HarpMessage, CancellationToken\)](#) , [AsyncDevice.Dispose\(\)](#) ,  
[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Methods

### ReadBrakeCurrentSetPointAsync(CancellationToken)

Asynchronously reads the contents of the BrakeCurrentSetPoint register.

```
public Task<ushort> ReadBrakeCurrentSetPointAsync(CancellationToken cancellationToken  
= default)
```

## Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#) <[ushort](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadEnableTorqueLimitAsync(CancellationToken)

Asynchronously reads the contents of the EnableTorqueLimit register.

```
public Task<EnableFlag> ReadEnableTorqueLimitAsync(CancellationToken cancellationToken  
= default)
```

## Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#) <[EnableFlag](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadEncoderAsync(CancellationToken)

Asynchronously reads the contents of the Encoder register.

```
public Task<int> ReadEncoderAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[int](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadResetTareSensorsAsync(CancellationToken)

Asynchronously reads the contents of the ResetTareSensors register.

```
public Task<Sensors> ReadResetTareSensorsAsync(CancellationToken cancellationToken  
= default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Sensors](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadSensorDataAsync(CancellationToken)

Asynchronously reads the contents of the SensorData register.

```
public Task<SensorDataPayload> ReadSensorDataAsync(CancellationToken cancellationToken  
= default)
```

## Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#) <[SensorDataPayload](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadSensorDataDispatchRateAsync(CancellationToken)

Asynchronously reads the contents of the SensorDataDispatchRate register.

```
public Task<ushort> ReadSensorDataDispatchRateAsync(CancellationToken cancellationToken  
= default)
```

## Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#) <[ushort](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadTareSensorsAsync(CancellationToken)

Asynchronously reads the contents of the TareSensors register.

```
public Task<Sensors> ReadTareSensorsAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Sensors](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadTimestampedBrakeCurrentSetPointAsync(CancellationToken)

Asynchronously reads the timestamped contents of the BrakeCurrentSetPoint register.

```
public Task<Timestamped<ushort>> ReadTimestampedBrakeCurrentSetPointAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<[ushort](#)>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedEnableTorqueLimitAsync(CancellationToken)

Asynchronously reads the timestamped contents of the EnableTorqueLimit register.

```
public Task<Timestamped<EnableFlag>> ReadTimestampedEnableTorqueLimitAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<[EnableFlag](#)>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedEncoderAsync(CancellationToken)

Asynchronously reads the timestamped contents of the Encoder register.

```
public Task<Timestamped<int>> ReadTimestampedEncoderAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<[int](#)>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedResetTareSensorsAsync(CancellationToken)

Asynchronously reads the timestamped contents of the ResetTareSensors register.

```
public Task<Timestamped<Sensors>> ReadTimestampedResetTareSensorsAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<[Sensors](#)>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedSensorDataAsync(CancellationToken)

Asynchronously reads the timestamped contents of the SensorData register.

```
public Task<Timestamped<SensorDataPayload>> ReadTimestampedSensorDataAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<[SensorDataPayload](#)>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedSensorDataDispatchRateAsync(CancellationToken)

Asynchronously reads the timestamped contents of the SensorDataDispatchRate register.

```
public Task<Timestamped<ushort>>
ReadTimestampedSensorDataDispatchRateAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<ushort>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedTareSensorsAsync(CancellationToken)

Asynchronously reads the timestamped contents of the TareSensors register.

```
public Task<Timestamped<Sensors>> ReadTimestampedTareSensorsAsync(CancellationToken
cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<Sensors>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedTorqueAsync(CancellationToken)

Asynchronously reads the timestamped contents of the Torque register.

```
public Task<Timestamped<short>> ReadTimestampedTorqueAsync(CancellationToken  
cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<short>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedTorqueLimitStateAsync(CancellationToken)

Asynchronously reads the timestamped contents of the TorqueLimitState register.

```
public Task<Timestamped<byte>> ReadTimestampedTorqueLimitStateAsync(CancellationToken  
cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<byte>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedTorqueLoadCurrentAsync(CancellationToken)

Asynchronously reads the timestamped contents of the TorqueLoadCurrent register.

```
public Task<Timestamped<short>> ReadTimestampedTorqueLoadCurrentAsync(CancellationToken cancellationToken = default)
```

## Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<short>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTorqueAsync(CancellationToken)

Asynchronously reads the contents of the Torque register.

```
public Task<short> ReadTorqueAsync(CancellationToken cancellationToken = default)
```

## Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<short>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadTorqueLimitStateAsync(CancellationToken)

Asynchronously reads the contents of the TorqueLimitState register.

```
public Task<byte> ReadTorqueLimitStateAsync(CancellationToken cancellationToken = default)
```

Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

Returns

[Task](#)<[byte](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadTorqueLoadCurrentAsync(CancellationToken)

Asynchronously reads the contents of the TorqueLoadCurrent register.

```
public Task<short> ReadTorqueLoadCurrentAsync(CancellationToken cancellationToken = default)
```

Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

Returns

[Task](#)<[short](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## WriteBrakeCurrentSetPointAsync(ushort, CancellationToken)

Asynchronously writes a value to the BrakeCurrentSetPoint register.

```
public Task WriteBrakeCurrentSetPointAsync(ushort value, CancellationToken cancellationToken = default)
```

## Parameters

### `value` [ushort](#)

The value to be stored in the register.

### `cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

### [Task](#)

The task object representing the asynchronous write operation.

## WriteEnableTorqueLimitAsync(EnableFlag, CancellationToken)

Asynchronously writes a value to the EnableTorqueLimit register.

```
public Task WriteEnableTorqueLimitAsync(EnableFlag value, CancellationToken cancellationToken = default)
```

## Parameters

### `value` [EnableFlag](#)

The value to be stored in the register.

### `cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

### [Task](#)

The task object representing the asynchronous write operation.

## WriteResetTareSensorsAsync(Sensors, CancellationToken)

Asynchronously writes a value to the ResetTareSensors register.

```
public Task WriteResetTareSensorsAsync(Sensors value, CancellationToken cancellationToken  
= default)
```

### Parameters

**value** [Sensors](#)

The value to be stored in the register.

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

### Returns

[Task](#)

The task object representing the asynchronous write operation.

## WriteSensorDataDispatchRateAsync(ushort, CancellationToken)

Asynchronously writes a value to the SensorDataDispatchRate register.

```
public Task WriteSensorDataDispatchRateAsync(ushort value, CancellationToken  
cancellationToken = default)
```

### Parameters

**value** [ushort](#)

The value to be stored in the register.

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

### Returns

## [Task](#)

The task object representing the asynchronous write operation.

# WriteTareSensorsAsync(Sensors, CancellationToken)

Asynchronously writes a value to the TareSensors register.

```
public Task WriteTareSensorsAsync(Sensors value, CancellationToken cancellationToken = default)
```

## Parameters

### [value Sensors](#)

The value to be stored in the register.

### [cancellationToken CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

## [Task](#)

The task object representing the asynchronous write operation.

# WriteTorqueLimitStateAsync(byte, CancellationToken)

Asynchronously writes a value to the TorqueLimitState register.

```
public Task WriteTorqueLimitStateAsync(byte value, CancellationToken cancellationToken = default)
```

## Parameters

### [value byte](#)

The value to be stored in the register.

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

Returns

[Task](#)

The task object representing the asynchronous write operation.

# Class BrakeCurrentSetPoint

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents a register that sets the raw value of the torque set-point to be applied to the treadmill. This value is cleared to 0 if torque\_limiting is enabled and triggered. Further writes in this condition return a WRITE\_ERROR.

```
public class BrakeCurrentSetPoint
```

## Inheritance

[object](#) ← BrakeCurrentSetPoint

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [BrakeCurrentSetPoint](#) register. This field is constant.

```
public const int Address = 37
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [BrakeCurrentSetPoint](#) register. This field is constant.

```
public const int RegisterLength = 1
```

## Field Value

[int↗](#)

## RegisterType

Represents the payload type of the [BrakeCurrentSetPoint](#) register. This field is constant.

```
public const PayloadType RegisterType = U16
```

## Field Value

[PayloadType↗](#)

## Methods

### FromPayload(MessageType, ushort)

Returns a Harp message for the [BrakeCurrentSetPoint](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, ushort value)
```

#### Parameters

**messageType** [MessageType↗](#)

The type of the Harp message.

**value** [ushort↗](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage↗](#)

A [HarpMessage↗](#) object for the [BrakeCurrentSetPoint](#) register with the specified message type and payload.

## FromPayload(double, MessageType, ushort)

Returns a timestamped Harp message for the [BrakeCurrentSetPoint](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
ushort value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [ushort](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [BrakeCurrentSetPoint](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [BrakeCurrentSetPoint](#) register messages.

```
public static ushort GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## [ushort](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [BrakeCurrentSetPoint](#) register messages.

```
public static Timestamped<ushort> GetTimestampedPayload(HarpMessage message)
```

### Parameters

`message` [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) < ushort >

A value representing the timestamped message payload.

# Class CreateBrakeCurrentSetPointPayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that creates a message payload that sets the raw value of the torque set-point to be applied to the treadmill. This value is cleared to 0 if torque\_limiting is enabled and triggered. Further writes in this condition return a WRITE\_ERROR.

```
public class CreateBrakeCurrentSetPointPayload
```

## Inheritance

[object](#) ← CreateBrakeCurrentSetPointPayload

## Derived

[CreateTimestampedBrakeCurrentSetPointPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## BrakeCurrentSetPoint

Gets or sets the value that sets the raw value of the torque set-point to be applied to the treadmill. This value is cleared to 0 if torque\_limiting is enabled and triggered. Further writes in this condition return a WRITE\_ERROR.

```
[Range(0, 65535)]  
public ushort BrakeCurrentSetPoint { get; set; }
```

## Property Value

[ushort](#)

# Methods

## GetMessage(MessageType)

Creates a message that sets the raw value of the torque set-point to be applied to the treadmill. This value is cleared to 0 if torque\_limiting is enabled and triggered. Further writes in this condition return a WRITE\_ERROR.

```
public HarpMessage GetMessage(MessageType messageType)
```

### Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

### Returns

[HarpMessage](#)

A new message for the BrakeCurrentSetPoint register.

## GetPayload()

Creates a message payload for the BrakeCurrentSetPoint register.

```
public ushort GetPayload()
```

### Returns

[ushort](#)

The created message payload value.

# Class CreateEnableTorqueLimitPayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that creates a message payload that enables(1)/Disables(0) the brake if the maximum torque sensor value is detected. This register will be enabled by default.

```
public class CreateEnableTorqueLimitPayload
```

## Inheritance

[object](#) ← CreateEnableTorqueLimitPayload

## Derived

[CreateTimestampedEnableTorqueLimitPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## EnableTorqueLimit

Gets or sets the value that enables(1)/Disables(0) the brake if the maximum torque sensor value is detected. This register will be enabled by default.

```
public EnableFlag EnableTorqueLimit { get; set; }
```

## Property Value

[EnableFlag](#)

# Methods

## GetMessage(MessageType)

Creates a message that enables(1)/Disables(0) the brake if the maximum torque sensor value is detected. This register will be enabled by default.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the EnableTorqueLimit register.

## GetPayload()

Creates a message payload for the EnableTorqueLimit register.

```
public EnableFlag GetPayload()
```

## Returns

[EnableFlag](#)

The created message payload value.

# Class CreateEncoderPayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that creates a message payload that contains the current accumulated number of ticks.

```
public class CreateEncoderPayload
```

## Inheritance

[object](#) ← CreateEncoderPayload

## Derived

[CreateTimestampedEncoderPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Encoder

Gets or sets the value that contains the current accumulated number of ticks.

```
public int Encoder { get; set; }
```

## Property Value

[int](#)

# Methods

## GetMessage(MessageType)

Creates a message that contains the current accumulated number of ticks.

```
public HarpMessage GetMessage(MessageType messageType)
```

Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

Returns

[HarpMessage](#)

A new message for the Encoder register.

## GetPayload()

Creates a message payload for the Encoder register.

```
public int GetPayload()
```

Returns

[int](#)

The created message payload value.

# Class CreateMessage

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator which creates standard message payloads for the Treadmill device.

```
public class CreateMessage : CreateMessageBuilder, IExpressionBuilder,  
ICustomTypeDescriptor, INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [CreateMessageBuilder](#) ← CreateMessage

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[CreateMessageBuilder.Build\(IEnumerable<Expression>\)](#), [CreateMessageBuilder.ArgumentRange](#),  
[CreateMessageBuilder.MessageType](#), [CreateMessageBuilder.Payload](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

## Constructors

### CreateMessage()

Initializes a new instance of the [CreateMessage](#) class.

```
public CreateMessage()
```

## See Also

[CreateEncoderPayload](#)

[CreateTorquePayload](#)

[CreateTorqueLoadCurrentPayload](#)

[CreateSensorDataPayload](#)

[CreateSensorDataDispatchRatePayload](#)

[CreateBrakeCurrentSetPointPayload](#)

[CreateTareSensorsPayload](#)

[CreateResetTareSensorsPayload](#)

[CreateEnableTorqueLimitPayload](#)

[CreateTorqueLimitStatePayload](#)

# Class CreateResetTareSensorsPayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that creates a message payload that removes the tare from the specified sensors.

```
public class CreateResetTareSensorsPayload
```

## Inheritance

[object](#) ← CreateResetTareSensorsPayload

## Derived

[CreateTimestampedResetTareSensorsPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Properties

### ResetTareSensors

Gets or sets the value that removes the tare from the specified sensors.

```
public Sensors ResetTareSensors { get; set; }
```

## Property Value

[Sensors](#)

## Methods

### GetMessage(MessageType)

Creates a message that removes the tare from the specified sensors.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the ResetTareSensors register.

## GetPayload()

Creates a message payload for the ResetTareSensors register.

```
public Sensors GetPayload()
```

## Returns

[Sensors](#)

The created message payload value.

# Class CreateSensorDataDispatchRatePayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that creates a message payload that value greater than 0 will enable the periodic dispatch of treadmill data events at the specified rate (sp/s).

```
public class CreateSensorDataDispatchRatePayload
```

## Inheritance

[object](#) ← CreateSensorDataDispatchRatePayload

## Derived

[CreateTimestampedSensorDataDispatchRatePayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## SensorDataDispatchRate

Gets or sets the value that value greater than 0 will enable the periodic dispatch of treadmill data events at the specified rate (sp/s).

```
[Range(0, 1000)]  
public ushort SensorDataDispatchRate { get; set; }
```

## Property Value

[ushort](#)

# Methods

## GetMessage(MessageType)

Creates a message that value greater than 0 will enable the periodic dispatch of treadmill data events at the specified rate (sp/s).

```
public HarpMessage GetMessage(MessageType messageType)
```

### Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

### Returns

[HarpMessage](#)

A new message for the SensorDataDispatchRate register.

## GetPayload()

Creates a message payload for the SensorDataDispatchRate register.

```
public ushort GetPayload()
```

### Returns

[ushort](#)

The created message payload value.

# Class CreateSensorDataPayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that creates a message payload that emits a periodic event containing the packaged treadmill data. [Encoder, Torque, TorqueLoadCurrent].

```
public class CreateSensorDataPayload
```

## Inheritance

[object](#) ← CreateSensorDataPayload

## Derived

[CreateTimestampedSensorDataPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Encoder

Gets or sets a value to write on payload member Encoder.

```
public int Encoder { get; set; }
```

## Property Value

[int](#)

## Torque

Gets or sets a value to write on payload member Torque.

```
public int Torque { get; set; }
```

Property Value

[int](#)

## TorqueLoadCurrent

Gets or sets a value to write on payload member TorqueLoadCurrent.

```
public int TorqueLoadCurrent { get; set; }
```

Property Value

[int](#)

## Methods

### GetMessage(MessageType)

Creates a message that emits a periodic event containing the packaged treadmill data. [Encoder, Torque, TorqueLoadCurrent].

```
public HarpMessage GetMessage(MessageType messageType)
```

Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

Returns

[HarpMessage](#)

A new message for the SensorData register.

## GetPayload()

Creates a message payload for the SensorData register.

```
public SensorDataPayload GetPayload()
```

Returns

[SensorDataPayload](#)

The created message payload value.

# Class CreateTareSensorsPayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that creates a message payload that tares the specified sensors.

```
public class CreateTareSensorsPayload
```

## Inheritance

[object](#) ← CreateTareSensorsPayload

## Derived

[CreateTimestampedTareSensorsPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## TareSensors

Gets or sets the value that tares the specified sensors.

```
public Sensors TareSensors { get; set; }
```

## Property Value

[Sensors](#)

# Methods

## GetMessage(MessageType)

Creates a message that tares the specified sensors.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the TareSensors register.

## GetPayload()

Creates a message payload for the TareSensors register.

```
public Sensors GetPayload()
```

## Returns

[Sensors](#)

The created message payload value.

# Class

## CreateTimestampedBrakeCurrentSetPointPayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that creates a timestamped message payload that sets the raw value of the torque set-point to be applied to the treadmill. This value is cleared to 0 if torque\_limiting is enabled and triggered. Further writes in this condition return a WRITE\_ERROR.

```
public class CreateTimestampedBrakeCurrentSetPointPayload :  
CreateBrakeCurrentSetPointPayload
```

### Inheritance

[object](#) ← [CreateBrakeCurrentSetPointPayload](#) ← CreateTimestampedBrakeCurrentSetPointPayload

### Inherited Members

[CreateBrakeCurrentSetPointPayload.BrakeCurrentSetPoint](#) ,  
[CreateBrakeCurrentSetPointPayload.GetPayload\(\)](#) ,  
[CreateBrakeCurrentSetPointPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that sets the raw value of the torque set-point to be applied to the treadmill. This value is cleared to 0 if torque\_limiting is enabled and triggered. Further writes in this condition return a WRITE\_ERROR.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

Returns

[HarpMessage](#)

A new timestamped message for the BrakeCurrentSetPoint register.

# Class

# CreateTimestampedEnableTorqueLimitPayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that creates a timestamped message payload that enables(1)/Disables(0) the brake if the maximum torque sensor value is detected. This register will be enabled by default.

```
public class CreateTimestampedEnableTorqueLimitPayload : CreateEnableTorqueLimitPayload
```

## Inheritance

[object](#) ← [CreateEnableTorqueLimitPayload](#) ← CreateTimestampedEnableTorqueLimitPayload

## Inherited Members

[CreateEnableTorqueLimitPayload.EnableTorqueLimit](#) , [CreateEnableTorqueLimitPayload.GetPayload\(\)](#) ,  
[CreateEnableTorqueLimitPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that enables(1)/Disables(0) the brake if the maximum torque sensor value is detected. This register will be enabled by default.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new timestamped message for the EnableTorqueLimit register.

# Class CreateTimestampedEncoderPayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that creates a timestamped message payload that contains the current accumulated number of ticks.

```
public class CreateTimestampedEncoderPayload : CreateEncoderPayload
```

## Inheritance

[object](#) ← [CreateEncoderPayload](#) ← CreateTimestampedEncoderPayload

## Inherited Members

[CreateEncoderPayload.Encoder](#) , [CreateEncoderPayload.GetPayload\(\)](#) ,  
[CreateEncoderPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that contains the current accumulated number of ticks.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

#### Returns

## [HarpMessage](#)

A new timestamped message for the Encoder register.

# Class CreateTimestampedResetTareSensorsPayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that creates a timestamped message payload that removes the tare from the specified sensors.

```
public class CreateTimestampedResetTareSensorsPayload : CreateResetTareSensorsPayload
```

## Inheritance

[object](#) ← [CreateResetTareSensorsPayload](#) ← CreateTimestampedResetTareSensorsPayload

## Inherited Members

[CreateResetTareSensorsPayload.ResetTareSensors](#) , [CreateResetTareSensorsPayload.GetPayload\(\)](#) ,  
[CreateResetTareSensorsPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that removes the tare from the specified sensors.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

### [HarpMessage](#)

A new timestamped message for the ResetTareSensors register.

# Class

# CreateTimestampedSensorDataDispatchRatePayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that creates a timestamped message payload that value greater than 0 will enable the periodic dispatch of treadmill data events at the specified rate (sp/s).

```
public class CreateTimestampedSensorDataDispatchRatePayload :  
CreateSensorDataDispatchRatePayload
```

## Inheritance

[object](#) ← [CreateSensorDataDispatchRatePayload](#) ←  
CreateTimestampedSensorDataDispatchRatePayload

## Inherited Members

[CreateSensorDataDispatchRatePayload.SensorDataDispatchRate](#) ,  
[CreateSensorDataDispatchRatePayload.GetPayload\(\)](#) ,  
[CreateSensorDataDispatchRatePayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that value greater than 0 will enable the periodic dispatch of treadmill data events at the specified rate (sp/s).

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

timestamp [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

Returns

[HarpMessage](#)

A new timestamped message for the SensorDataDispatchRate register.

# Class CreateTimestampedSensorDataPayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that creates a timestamped message payload that emits a periodic event containing the packaged treadmill data. [Encoder, Torque, TorqueLoadCurrent].

```
public class CreateTimestampedSensorDataPayload : CreateSensorDataPayload
```

## Inheritance

[object](#) ← [CreateSensorDataPayload](#) ← CreateTimestampedSensorDataPayload

## Inherited Members

[CreateSensorDataPayload.Encoder](#) , [CreateSensorDataPayload.Torque](#) ,  
[CreateSensorDataPayload.TorqueLoadCurrent](#) , [CreateSensorDataPayload.GetPayload\(\)](#) ,  
[CreateSensorDataPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that emits a periodic event containing the packaged treadmill data. [Encoder, Torque, TorqueLoadCurrent].

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new timestamped message for the SensorData register.

# Class CreateTimestampedTareSensorsPayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that creates a timestamped message payload that tares the specified sensors.

```
public class CreateTimestampedTareSensorsPayload : CreateTareSensorsPayload
```

## Inheritance

[object](#) ← [CreateTareSensorsPayload](#) ← CreateTimestampedTareSensorsPayload

## Inherited Members

[CreateTareSensorsPayload.TareSensors](#) , [CreateTareSensorsPayload.GetPayload\(\)](#) ,  
[CreateTareSensorsPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that tares the specified sensors.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

#### Returns

[HarpMessage](#)

A new timestamped message for the TareSensors register.

# Class

# CreateTimestampedTorqueLimitStatePayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that creates a timestamped message payload that a value greater than 1 indicates that the torque limit has been triggered and the brake setpoint will be cleared. Writing a value of 0 will clear the torque limit state and re-enable the brake.

```
public class CreateTimestampedTorqueLimitStatePayload : CreateTorqueLimitStatePayload
```

## Inheritance

[object](#) ← [CreateTorqueLimitStatePayload](#) ← CreateTimestampedTorqueLimitStatePayload

## Inherited Members

[CreateTorqueLimitStatePayload.TorqueLimitState](#) , [CreateTorqueLimitStatePayload.GetPayload\(\)](#) ,  
[CreateTorqueLimitStatePayload GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that a value greater than 1 indicates that the torque limit has been triggered and the brake setpoint will be cleared. Writing a value of 0 will clear the torque limit state and re-enable the brake.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

Returns

[HarpMessage](#) ↗

A new timestamped message for the TorqueLimitState register.

# Class

# CreateTimestampedTorqueLoadCurrentPayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that creates a timestamped message payload that contains the current output current applied to the variable torque load.

```
public class CreateTimestampedTorqueLoadCurrentPayload : CreateTorqueLoadCurrentPayload
```

## Inheritance

[object](#) ← [CreateTorqueLoadCurrentPayload](#) ← CreateTimestampedTorqueLoadCurrentPayload

## Inherited Members

[CreateTorqueLoadCurrentPayload.TorqueLoadCurrent](#) , [CreateTorqueLoadCurrentPayload.GetPayload\(\)](#) ,  
[CreateTorqueLoadCurrentPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that contains the current output current applied to the variable torque load.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new timestamped message for the TorqueLoadCurrent register.

# Class CreateTimestampedTorquePayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that creates a timestamped message payload that contains the current torque value.

```
public class CreateTimestampedTorquePayload : CreateTorquePayload
```

## Inheritance

[object](#) ← [CreateTorquePayload](#) ← CreateTimestampedTorquePayload

## Inherited Members

[CreateTorquePayload.Torque](#) , [CreateTorquePayload.GetPayload\(\)](#) ,  
[CreateTorquePayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that contains the current torque value.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

#### Returns

## [HarpMessage](#)

A new timestamped message for the Torque register.

# Class CreateTorqueLimitStatePayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that creates a message payload that a value greater than 1 indicates that the torque limit has been triggered and the brake setpoint will be cleared. Writing a value of 0 will clear the torque limit state and re-enable the brake.

```
public class CreateTorqueLimitStatePayload
```

## Inheritance

[object](#) ← CreateTorqueLimitStatePayload

## Derived

[CreateTimestampedTorqueLimitStatePayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Properties

### TorqueLimitState

Gets or sets the value that a value greater than 1 indicates that the torque limit has been triggered and the brake setpoint will be cleared. Writing a value of 0 will clear the torque limit state and re-enable the brake.

```
public byte TorqueLimitState { get; set; }
```

## Property Value

[byte](#)

## Methods

## GetMessage(MessageType)

Creates a message that a value greater than 1 indicates that the torque limit has been triggered and the brake setpoint will be cleared. Writing a value of 0 will clear the torque limit state and re-enable the brake.

```
public HarpMessage GetMessage(MessageType messageType)
```

### Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

### Returns

[HarpMessage](#)

A new message for the TorqueLimitState register.

## GetPayload()

Creates a message payload for the TorqueLimitState register.

```
public byte GetPayload()
```

### Returns

[byte](#)

The created message payload value.

# Class CreateTorqueLoadCurrentPayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that creates a message payload that contains the current output current applied to the variable torque load.

```
public class CreateTorqueLoadCurrentPayload
```

## Inheritance

[object](#) ← CreateTorqueLoadCurrentPayload

## Derived

[CreateTimestampedTorqueLoadCurrentPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## TorqueLoadCurrent

Gets or sets the value that contains the current output current applied to the variable torque load.

```
public short TorqueLoadCurrent { get; set; }
```

## Property Value

[short](#)

# Methods

## GetMessage(MessageType)

Creates a message that contains the current output current applied to the variable torque load.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the TorqueLoadCurrent register.

## GetPayload()

Creates a message payload for the TorqueLoadCurrent register.

```
public short GetPayload()
```

## Returns

[short](#)

The created message payload value.

# Class CreateTorquePayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that creates a message payload that contains the current torque value.

```
public class CreateTorquePayload
```

## Inheritance

[object](#) ← CreateTorquePayload

## Derived

[CreateTimestampedTorquePayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Torque

Gets or sets the value that contains the current torque value.

```
public short Torque { get; set; }
```

## Property Value

[short](#)

# Methods

## GetMessage(MessageType)

Creates a message that contains the current torque value.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the Torque register.

## GetPayload()

Creates a message payload for the Torque register.

```
public short GetPayload()
```

## Returns

[short](#)

The created message payload value.

# Class Device

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an observable source of messages from the Harp device connected at the specified serial port.

```
[Combinator(MethodName = "Generate")]
[WorkflowElementCategory(ElementCategory.Source)]
public class Device : Device, INamedElement
```

## Inheritance

[object](#) ← [Source](#)<[HarpMessage](#)> ← [Device](#) ← [Device](#)

## Implements

[INamedElement](#)

## Inherited Members

[Device.Generate\(\)](#) , [Device.Generate\(IObservable<HarpMessage>\)](#) , [Device.OperationMode](#) ,  
[Device.OperationLed](#) , [Device.DumpRegisters](#) , [Device.VisualIndicators](#) , [Device.Heartbeat](#) ,  
[Device.IgnoreErrors](#) , [Device.PortName](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

# Constructors

## Device()

Initializes a new instance of the [Device](#) class.

```
public Device()
```

# Fields

## Metadata

Gets the contents of the metadata file describing the [AllenNeuralDynamics.Treadmill](#) device registers.

```
public static readonly string Metadata
```

Field Value

[string](#)

## WhoAmI

Represents the unique identity class of the [AllenNeuralDynamics.Treadmill](#) device. This field is constant.

```
public const int WhoAmI = 1402
```

Field Value

[int](#)

## Properties

### RegisterMap

Gets a read-only mapping from address to register type.

```
public static IReadOnlyDictionary<int, Type> RegisterMap { get; }
```

Property Value

[IReadOnlyDictionary](#)<[int](#), [Type](#)>

## Methods

### CreateAsync(string)

Initializes a new instance of the asynchronous API to configure and interface with Treadmill devices on the specified serial port.

```
public static Task<AsyncDevice> CreateAsync(string portName)
```

## Parameters

portName [string](#)

The name of the serial port used to communicate with the Harp device.

## Returns

[Task](#) <[AsyncDevice](#)>

A task that represents the asynchronous initialization operation. The value of the [Result](#) parameter contains a new instance of the [AsyncDevice](#) class.

# Class EnableTorqueLimit

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents a register that enables(1)/Disables(0) the brake if the maximum torque sensor value is detected. This register will be enabled by default.

```
public class EnableTorqueLimit
```

## Inheritance

[object](#) ← EnableTorqueLimit

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [EnableTorqueLimit](#) register. This field is constant.

```
public const int Address = 40
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [EnableTorqueLimit](#) register. This field is constant.

```
public const int RegisterLength = 1
```

## Field Value

[int↗](#)

## RegisterType

Represents the payload type of the [EnableTorqueLimit](#) register. This field is constant.

```
public const PayloadType RegisterType = U8
```

## Field Value

[PayloadType↗](#)

## Methods

### FromPayload(MessageType, EnableFlag)

Returns a Harp message for the [EnableTorqueLimit](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, EnableFlag value)
```

#### Parameters

**messageType** [MessageType↗](#)

The type of the Harp message.

**value** [EnableFlag↗](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage↗](#)

A [HarpMessage↗](#) object for the [EnableTorqueLimit](#) register with the specified message type and payload.

## FromPayload(double, MessageType, EnableFlag)

Returns a timestamped Harp message for the [EnableTorqueLimit](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
EnableFlag value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [EnableFlag](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [EnableTorqueLimit](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [EnableTorqueLimit](#) register messages.

```
public static EnableFlag GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## [EnableFlag](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [EnableTorqueLimit](#) register messages.

```
public static Timestamped<EnableFlag> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <[EnableFlag](#)>

A value representing the timestamped message payload.

# Class Encoder

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents a register that contains the current accumulated number of ticks.

```
public class Encoder
```

## Inheritance

[object](#) ← Encoder

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [Encoder](#) register. This field is constant.

```
public const int Address = 32
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [Encoder](#) register. This field is constant.

```
public const int RegisterLength = 1
```

### Field Value

## .RegisterType

Represents the payload type of the [Encoder](#) register. This field is constant.

```
public const PayloadType RegisterType = S32
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, int)

Returns a Harp message for the [Encoder](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, int value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [int](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [Encoder](#) register with the specified message type and payload.

### FromPayload(double, MessageType, int)

Returns a timestamped Harp message for the [Encoder](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType, int value)
```

Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [int](#)

The value to be stored in the message payload.

Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [Encoder](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [Encoder](#) register messages.

```
public static int GetPayload(HarpMessage message)
```

Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

Returns

[int](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [Encoder](#) register messages.

```
public static Timestamped<int> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#)<int>

A value representing the timestamped message payload.

# Class FilterRegister

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that filters register-specific messages reported by the [AllenNeuralDynamics.Treadmill](#) device.

```
public class FilterRegister : FilterRegisterBuilder, IExpressionBuilder,  
ICustomTypeDescriptor, INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [RegisterCombinatorBuilder](#) ←  
[FilterRegisterBuilder](#) ← FilterRegister

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[FilterRegisterBuilder.FilterType](#), [FilterRegisterBuilder.Register](#),  
[RegisterCombinatorBuilder.Build\(IEnumerable<Expression>\)](#),  
[RegisterCombinatorBuilder.ArgumentRange](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

## Constructors

### FilterRegister()

Initializes a new instance of the [FilterRegister](#) class.

```
public FilterRegister()
```

## See Also

[Encoder](#)

[Torque](#)

[TorqueLoadCurrent](#)

[SensorData](#)

[SensorDataDispatchRate](#)

[BrakeCurrentSetPoint](#)

[TareSensors](#)

[ResetTareSensors](#)

[EnableTorqueLimit](#)

[TorqueLimitState](#)

# Class Format

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator which formats a sequence of values as specific Treadmill register messages.

```
public class Format : FormatBuilder, IExpressionBuilder, ICustomTypeDescriptor,  
INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [FormatBuilder](#) ← Format

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[FormatBuilder.Build\(IEnumerable<Expression>\)](#), [FormatBuilder.ArgumentRange](#),  
[FormatBuilder.MessageType](#), [FormatBuilder.Register](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

## Constructors

### Format()

Initializes a new instance of the [Format](#) class.

```
public Format()
```

## See Also

[Encoder](#)

[Torque](#)

[TorqueLoadCurrent](#)

[SensorData](#)

[SensorDataDispatchRate](#)

[BrakeCurrentSetPoint](#)

[TareSensors](#)

[ResetTareSensors](#)

[EnableTorqueLimit](#)

[TorqueLimitState](#)

# Class GetMetadata

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that returns the contents of the metadata file describing the [AllenNeuralDynamics.Treadmill](#) device registers.

```
public class GetMetadata : Source<string>
```

## Inheritance

[object](#) ← [Source](#)<[string](#)> ← GetMetadata

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Methods

### Generate()

Returns an observable sequence with the contents of the metadata file describing the [AllenNeuralDynamics.Treadmill](#) device registers.

```
public override IObservable<string> Generate()
```

## Returns

[IObservable](#)<[string](#)>

A sequence with a single [string](#) object representing the contents of the metadata file.

# Class GroupByRegister

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator that groups the sequence of [AllenNeuralDynamics.Treadmill](#)" messages by register type.

```
public class GroupByRegister : Combinator<HarpMessage, IGroupedObservable<Type, HarpMessage>>
```

## Inheritance

```
object ↗ ← Combinator ↗ <HarpMessage ↗, IGroupedObservable ↗ <Type ↗, HarpMessage ↗>> ← GroupByRegister
```

## Inherited Members

```
Combinator<HarpMessage, IGroupedObservable<Type, HarpMessage>>.Process(I Observable<HarpMessage>) ↗ ,  
object.ToString() ↗ , object.Equals(object) ↗ , object.Equals(object, object) ↗ ,  
object.ReferenceEquals(object, object) ↗ , object.GetHashCode() ↗ , object.GetType() ↗ ,  
object.MemberwiseClone() ↗
```

## Methods

### Process(I Observable<HarpMessage>)

Groups an observable sequence of [AllenNeuralDynamics.Treadmill](#) messages by register type.

```
public override IObservable<IGroupedObservable<Type, HarpMessage>>  
Process(IObservable<HarpMessage> source)
```

## Parameters

source [IObservable](#)<HarpMessage>

The sequence of Harp device messages.

## Returns

[IObservable](#) < [IGroupedObservable](#) < [Type](#), [HarpMessage](#) >>

A sequence of observable groups, each of which corresponds to a unique [AllenNeuralDynamics.Treadmill](#) register.

# Class Parse

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents an operator which filters and selects specific messages reported by the Treadmill device.

```
public class Parse : ParseBuilder, IExpressionBuilder, ICustomTypeDescriptor, INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [RegisterCombinatorBuilder](#) ← [ParseBuilder](#) ← Parse

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[ParseBuilder.Register](#), [RegisterCombinatorBuilder.Build\(IEnumerable<Expression>\)](#),  
[RegisterCombinatorBuilder.ArgumentRange](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

## Constructors

### Parse()

Initializes a new instance of the [Parse](#) class.

```
public Parse()
```

## See Also

[Encoder](#)

[Torque](#)

[TorqueLoadCurrent](#)

[SensorData](#)

[SensorDataDispatchRate](#)

[BrakeCurrentSetPoint](#)

[TareSensors](#)

[ResetTareSensors](#)

[EnableTorqueLimit](#)

[TorqueLimitState](#)

# Class ResetTareSensors

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents a register that removes the tare from the specified sensors.

```
public class ResetTareSensors
```

## Inheritance

[object](#) ← ResetTareSensors

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [ResetTareSensors](#) register. This field is constant.

```
public const int Address = 39
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [ResetTareSensors](#) register. This field is constant.

```
public const int RegisterLength = 1
```

### Field Value

## .RegisterType

Represents the payload type of the [ResetTareSensors](#) register. This field is constant.

```
public const PayloadType RegisterType = U8
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, Sensors)

Returns a Harp message for the [ResetTareSensors](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, Sensors value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [Sensors](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [ResetTareSensors](#) register with the specified message type and payload.

## FromPayload(double, MessageType, Sensors)

Returns a timestamped Harp message for the [ResetTareSensors](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
Sensors value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [Sensors](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [ResetTareSensors](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [ResetTareSensors](#) register messages.

```
public static Sensors GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## [Sensors](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [ResetTareSensors](#) register messages.

```
public static Timestamped<Sensors> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <[Sensors](#)>

A value representing the timestamped message payload.

# Class SensorData

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents a register that emits a periodic event containing the packaged treadmill data. [Encoder, Torque, TorqueLoadCurrent].

```
public class SensorData
```

## Inheritance

[object](#) ← SensorData

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [SensorData](#) register. This field is constant.

```
public const int Address = 35
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [SensorData](#) register. This field is constant.

```
public const int RegisterLength = 3
```

## Field Value

[int](#)

## RegisterType

Represents the payload type of the [SensorData](#) register. This field is constant.

```
public const PayloadType RegisterType = S32
```

## Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, SensorDataPayload)

Returns a Harp message for the [SensorData](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, SensorDataPayload value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [SensorDataPayload](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [SensorData](#) register with the specified message type and payload.

## FromPayload(double, MessageType, SensorDataPayload)

Returns a timestamped Harp message for the [SensorData](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
SensorDataPayload value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [SensorDataPayload](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [SensorData](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [SensorData](#) register messages.

```
public static SensorDataPayload GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## [SensorDataPayload](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [SensorData](#) register messages.

```
public static Timestamped<SensorDataPayload> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <[SensorDataPayload](#)>

A value representing the timestamped message payload.

# Class SensorDataDispatchRate

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents a register that value greater than 0 will enable the periodic dispatch of treadmill data events at the specified rate (sp/s).

```
public class SensorDataDispatchRate
```

## Inheritance

[object](#) ← SensorDataDispatchRate

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [SensorDataDispatchRate](#) register. This field is constant.

```
public const int Address = 36
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [SensorDataDispatchRate](#) register. This field is constant.

```
public const int RegisterLength = 1
```

## Field Value

[int↗](#)

## RegisterType

Represents the payload type of the [SensorDataDispatchRate](#) register. This field is constant.

```
public const PayloadType RegisterType = U16
```

## Field Value

[PayloadType↗](#)

## Methods

### FromPayload(MessageType, ushort)

Returns a Harp message for the [SensorDataDispatchRate](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, ushort value)
```

#### Parameters

**messageType** [MessageType↗](#)

The type of the Harp message.

**value** [ushort↗](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage↗](#)

A [HarpMessage↗](#) object for the [SensorDataDispatchRate](#) register with the specified message type and payload.

## FromPayload(double, MessageType, ushort)

Returns a timestamped Harp message for the [SensorDataDispatchRate](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
ushort value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [ushort](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [SensorDataDispatchRate](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [SensorDataDispatchRate](#) register messages.

```
public static ushort GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## [ushort](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [SensorDataDispatchRate](#) register messages.

```
public static Timestamped<ushort> GetTimestampedPayload(HarpMessage message)
```

### Parameters

`message` [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <[ushort](#)>

A value representing the timestamped message payload.

# Struct SensorDataPayload

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents the payload of the SensorData register.

```
public struct SensorDataPayload
```

## Inherited Members

[ValueType.Equals\(object\)](#) , [ValueType.GetHashCode\(\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetType\(\)](#)

## Constructors

### SensorDataPayload(int, int, int)

Initializes a new instance of the [SensorDataPayload](#) structure.

```
public SensorDataPayload(int encoder, int torque, int torqueLoadCurrent)
```

## Parameters

encoder [int](#)

torque [int](#)

torqueLoadCurrent [int](#)

## Fields

### Encoder

```
public int Encoder
```

## Field Value

[int↗](#)

## Torque

`public int Torque`

### Field Value

[int↗](#)

## TorqueLoadCurrent

`public int TorqueLoadCurrent`

### Field Value

[int↗](#)

## Methods

### ToString()

Returns a [string↗](#) that represents the payload of the SensorData register.

`public override string ToString()`

### Returns

[string↗](#)

A [string↗](#) that represents the payload of the SensorData register.

# Enum Sensors

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Available sensors.

```
[Flags]
public enum Sensors : byte
```

## Fields

BrakeCurrent = 4

Encoder = 1

None = 0

Torque = 2

# Class TareSensors

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents a register that tares the specified sensors.

```
public class TareSensors
```

## Inheritance

[object](#) ← TareSensors

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [TareSensors](#) register. This field is constant.

```
public const int Address = 38
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [TareSensors](#) register. This field is constant.

```
public const int RegisterLength = 1
```

### Field Value

## .RegisterType

Represents the payload type of the [TareSensors](#) register. This field is constant.

```
public const PayloadType RegisterType = U8
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, Sensors)

Returns a Harp message for the [TareSensors](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, Sensors value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [Sensors](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [TareSensors](#) register with the specified message type and payload.

### FromPayload(double, MessageType, Sensors)

Returns a timestamped Harp message for the [TareSensors](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
Sensors value)
```

Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [Sensors](#)

The value to be stored in the message payload.

Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [TareSensors](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [TareSensors](#) register messages.

```
public static Sensors GetPayload(HarpMessage message)
```

Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

Returns

[Sensors](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [TareSensors](#) register messages.

```
public static Timestamped<Sensors> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#)<[Sensors](#)>

A value representing the timestamped message payload.

# Class TimestampedBrakeCurrentSetPoint

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Provides methods for manipulating timestamped messages from the BrakeCurrentSetPoint register.

```
public class TimestampedBrakeCurrentSetPoint
```

## Inheritance

[object](#) ← TimestampedBrakeCurrentSetPoint

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [BrakeCurrentSetPoint](#) register. This field is constant.

```
public const int Address = 37
```

### Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [BrakeCurrentSetPoint](#) register messages.

```
public static Timestamped<ushort> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[ushort](#)>

A value representing the timestamped message payload.

## See Also

[BrakeCurrentSetPoint](#)

# Class TimestampedEnableTorqueLimit

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Provides methods for manipulating timestamped messages from the EnableTorqueLimit register.

```
public class TimestampedEnableTorqueLimit
```

## Inheritance

[object](#) ← TimestampedEnableTorqueLimit

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [EnableTorqueLimit](#) register. This field is constant.

```
public const int Address = 40
```

## Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [EnableTorqueLimit](#) register messages.

```
public static Timestamped<EnableFlag> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#)<[EnableFlag](#)>

A value representing the timestamped message payload.

## See Also

[EnableTorqueLimit](#)

# Class TimestampedEncoder

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Provides methods for manipulating timestamped messages from the Encoder register.

```
public class TimestampedEncoder
```

## Inheritance

[object](#) ← TimestampedEncoder

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [Encoder](#) register. This field is constant.

```
public const int Address = 32
```

## Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [Encoder](#) register messages.

```
public static Timestamped<int> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#)<[int](#)>

A value representing the timestamped message payload.

## See Also

[Encoder](#)

# Class TimestampedResetTareSensors

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Provides methods for manipulating timestamped messages from the ResetTareSensors register.

```
public class TimestampedResetTareSensors
```

## Inheritance

[object](#) ← TimestampedResetTareSensors

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [ResetTareSensors](#) register. This field is constant.

```
public const int Address = 39
```

### Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [ResetTareSensors](#) register messages.

```
public static Timestamped<Sensors> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[Sensors](#)>

A value representing the timestamped message payload.

## See Also

[ResetTareSensors](#)

# Class TimestampedSensorData

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Provides methods for manipulating timestamped messages from the SensorData register.

```
public class TimestampedSensorData
```

## Inheritance

[object](#) ← TimestampedSensorData

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [SensorData](#) register. This field is constant.

```
public const int Address = 35
```

### Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [SensorData](#) register messages.

```
public static Timestamped<SensorDataPayload> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[SensorDataPayload](#)>

A value representing the timestamped message payload.

## See Also

[SensorData](#)

# Class TimestampedSensorDataDispatchRate

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Provides methods for manipulating timestamped messages from the SensorDataDispatchRate register.

```
public class TimestampedSensorDataDispatchRate
```

## Inheritance

[object](#) ← TimestampedSensorDataDispatchRate

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [SensorDataDispatchRate](#) register. This field is constant.

```
public const int Address = 36
```

## Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [SensorDataDispatchRate](#) register messages.

```
public static Timestamped<ushort> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[ushort](#)>

A value representing the timestamped message payload.

## See Also

[SensorDataDispatchRate](#)

# Class TimestampedTareSensors

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Provides methods for manipulating timestamped messages from the TareSensors register.

```
public class TimestampedTareSensors
```

## Inheritance

[object](#) ← TimestampedTareSensors

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [TareSensors](#) register. This field is constant.

```
public const int Address = 38
```

## Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [TareSensors](#) register messages.

```
public static Timestamped<Sensors> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#)<[Sensors](#)>

A value representing the timestamped message payload.

## See Also

[TareSensors](#)

# Class TimestampedTorque

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Provides methods for manipulating timestamped messages from the Torque register.

```
public class TimestampedTorque
```

## Inheritance

[object](#) ← TimestampedTorque

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [Torque](#) register. This field is constant.

```
public const int Address = 33
```

### Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [Torque](#) register messages.

```
public static Timestamped<short> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#)<[short](#)>

A value representing the timestamped message payload.

## See Also

[Torque](#)

# Class TimestampedTorqueLimitState

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Provides methods for manipulating timestamped messages from the TorqueLimitState register.

```
public class TimestampedTorqueLimitState
```

## Inheritance

[object](#) ← TimestampedTorqueLimitState

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [TorqueLimitState](#) register. This field is constant.

```
public const int Address = 41
```

### Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [TorqueLimitState](#) register messages.

```
public static Timestamped<byte> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#)<[byte](#)>

A value representing the timestamped message payload.

## See Also

[TorqueLimitState](#)

# Class TimestampedTorqueLoadCurrent

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Provides methods for manipulating timestamped messages from the TorqueLoadCurrent register.

```
public class TimestampedTorqueLoadCurrent
```

## Inheritance

[object](#) ← TimestampedTorqueLoadCurrent

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [TorqueLoadCurrent](#) register. This field is constant.

```
public const int Address = 34
```

### Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [TorqueLoadCurrent](#) register messages.

```
public static Timestamped<short> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#)<[short](#)>

A value representing the timestamped message payload.

## See Also

[TorqueLoadCurrent](#)

# Class Torque

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents a register that contains the current torque value.

```
public class Torque
```

## Inheritance

[object](#) ← Torque

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [Torque](#) register. This field is constant.

```
public const int Address = 33
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [Torque](#) register. This field is constant.

```
public const int RegisterLength = 1
```

### Field Value

## .RegisterType

Represents the payload type of the [Torque](#) register. This field is constant.

```
public const PayloadType RegisterType = S16
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, short)

Returns a Harp message for the [Torque](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, short value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [short](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [Torque](#) register with the specified message type and payload.

### FromPayload(double, MessageType, short)

Returns a timestamped Harp message for the [Torque](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
short value)
```

Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [short](#)

The value to be stored in the message payload.

Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [Torque](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [Torque](#) register messages.

```
public static short GetPayload(HarpMessage message)
```

Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

Returns

[short](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [Torque](#) register messages.

```
public static Timestamped<short> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <short>

A value representing the timestamped message payload.

# Class TorqueLimitState

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents a register that a value greater than 1 indicates that the torque limit has been triggered and the brake setpoint will be cleared. Writing a value of 0 will clear the torque limit state and re-enable the brake.

```
public class TorqueLimitState
```

## Inheritance

[object](#) ← TorqueLimitState

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Fields

## Address

Represents the address of the [TorqueLimitState](#) register. This field is constant.

```
public const int Address = 41
```

## Field Value

[int](#)

## RegisterLength

Represents the length of the [TorqueLimitState](#) register. This field is constant.

```
public const int RegisterLength = 1
```

## Field Value

[int↗](#)

## RegisterType

Represents the payload type of the [TorqueLimitState](#) register. This field is constant.

```
public const PayloadType RegisterType = U8
```

## Field Value

[PayloadType↗](#)

## Methods

### FromPayload(MessageType, byte)

Returns a Harp message for the [TorqueLimitState](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, byte value)
```

#### Parameters

**messageType** [MessageType↗](#)

The type of the Harp message.

**value** [byte↗](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage↗](#)

A [HarpMessage↗](#) object for the [TorqueLimitState](#) register with the specified message type and payload.

## FromPayload(double, MessageType, byte)

Returns a timestamped Harp message for the [TorqueLimitState](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType, byte value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [byte](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [TorqueLimitState](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [TorqueLimitState](#) register messages.

```
public static byte GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## [byte](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [TorqueLimitState](#) register messages.

```
public static Timestamped<byte> GetTimestampedPayload(HarpMessage message)
```

### Parameters

[message](#) [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <[byte](#)>

A value representing the timestamped message payload.

# Class TorqueLoadCurrent

Namespace: [AllenNeuralDynamics.Treadmill](#)

Assembly: AllenNeuralDynamics.Treadmill.dll

Represents a register that contains the current output current applied to the variable torque load.

```
public class TorqueLoadCurrent
```

## Inheritance

[object](#) ← TorqueLoadCurrent

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [TorqueLoadCurrent](#) register. This field is constant.

```
public const int Address = 34
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [TorqueLoadCurrent](#) register. This field is constant.

```
public const int RegisterLength = 1
```

### Field Value

## .RegisterType

Represents the payload type of the [TorqueLoadCurrent](#) register. This field is constant.

```
public const PayloadType RegisterType = S16
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, short)

Returns a Harp message for the [TorqueLoadCurrent](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, short value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [short](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [TorqueLoadCurrent](#) register with the specified message type and payload.

## FromPayload(double, MessageType, short)

Returns a timestamped Harp message for the [TorqueLoadCurrent](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
short value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [short](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [TorqueLoadCurrent](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [TorqueLoadCurrent](#) register messages.

```
public static short GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## [short](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [TorqueLoadCurrent](#) register messages.

```
public static Timestamped<short> GetTimestampedPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <short>

A value representing the timestamped message payload.

# Namespace AllenNeuralDynamics.Version Control

## Classes

### [CreateRepository](#)

Represents an operator that populates a LibGit2Sharp.Repository object from a target folder path

### [IsRepositoryClean](#)

Represents an operator that asserts if a LibGit2Sharp.Repository object is clean and without untracked local changes.

# Class CreateRepository

Namespace: [AllenNeuralDynamics.VersionControl](#)

Assembly: AllenNeuralDynamics.VersionControl.dll

Represents an operator that populates a LibGit2Sharp.Repository object from a target folder path

```
public class CreateRepository : Source<Repository>
```

## Inheritance

[object](#) ← [Source](#)<Repository> ← CreateRepository

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Properties

### Path

Gets or sets path for the targeted repository.

```
public string Path { get; set; }
```

### Property Value

[string](#)

## Methods

### Generate()

Generates an observable with a single Repository object from a given root path.

```
public override IObservable<Repository> Generate()
```

## Returns

[IObservable](#) ↗ <Repository>

A sequence of LibGit2Sharp.Repository objects representing a git repository.

# Class IsRepositoryClean

Namespace: [AllenNeuralDynamics.VersionControl](#)

Assembly: AllenNeuralDynamics.VersionControl.dll

Represents an operator that asserts if a LibGit2Sharp.Repository object is clean and without untracked local changes.

```
public class IsRepositoryClean : Transform<Repository, bool>
```

## Inheritance

[object](#) ← [Combinator](#)<Repository, [bool](#)> ← [Transform](#)<Repository, [bool](#)> ← IsRepositoryClean

## Inherited Members

[Combinator<Repository, bool>.Process\(IObservable<Repository>\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Properties

### IgnoreUntracked

Optionally ignores untracked changes when determining if a repository is clean.

```
public bool IgnoreUntracked { get; set; }
```

## Property Value

[bool](#)

## Methods

### Process(I Observable<Repository>)

Emits a sequence of values indicating whether the repository is clean or not.

```
public override IObservable<bool> Process(IObservable<Repository> source)
```

## Parameters

source [IObservable](#)<Repository>

## Returns

[IObservable](#)<[bool](#)>

A sequence of [bool](#) values with assertion outcome.

# Namespace AllenNeuralDynamics.WhiteRabbit

## Classes

### [AsyncDevice](#)

Represents an asynchronous API to configure and interface with WhiteRabbit devices.

### [AuxPortBaudRate](#)

Represents a register that the baud rate, in bps, of the auxiliary port when in HarpClock mode.

### [AuxPortMode](#)

Represents a register that the function of the auxiliary port.

### [ConnectedDevices](#)

Represents a register that the currently connected output channels. An event will be generated when any of the channels are connected or disconnected.

### [Counter](#)

Represents a register that the counter value. This value is incremented at the frequency specified by CounterFrequencyHz. Write to force a counter value.

### [CounterFrequencyHz](#)

Represents a register that the frequency at which the counter is incremented. A value of 0 disables the counter.

### [CreateAuxPortBaudRatePayload](#)

Represents an operator that creates a message payload that the baud rate, in bps, of the auxiliary port when in HarpClock mode.

### [CreateAuxPortModePayload](#)

Represents an operator that creates a message payload that the function of the auxiliary port.

### [CreateConnectedDevicesPayload](#)

Represents an operator that creates a message payload that the currently connected output channels. An event will be generated when any of the channels are connected or disconnected.

### [CreateCounterFrequencyHzPayload](#)

Represents an operator that creates a message payload that the frequency at which the counter is incremented. A value of 0 disables the counter.

### [CreateCounterPayload](#)

Represents an operator that creates a message payload that the counter value. This value is incremented at the frequency specified by CounterFrequencyHz. Write to force a counter value.

## [CreateMessage](#)

Represents an operator which creates standard message payloads for the WhiteRabbit device.

## [CreateTimestampedAuxPortBaudRatePayload](#)

Represents an operator that creates a timestamped message payload that the baud rate, in bps, of the auxiliary port when in HarpClock mode.

## [CreateTimestampedAuxPortModePayload](#)

Represents an operator that creates a timestamped message payload that the function of the auxiliary port.

## [CreateTimestampedConnectedDevicesPayload](#)

Represents an operator that creates a timestamped message payload that the currently connected output channels. An event will be generated when any of the channels are connected or disconnected.

## [CreateTimestampedCounterFrequencyHzPayload](#)

Represents an operator that creates a timestamped message payload that the frequency at which the counter is incremented. A value of 0 disables the counter.

## [CreateTimestampedCounterPayload](#)

Represents an operator that creates a timestamped message payload that the counter value. This value is incremented at the frequency specified by CounterFrequencyHz. Write to force a counter value.

## [Device](#)

Represents an observable source of messages from the Harp device connected at the specified serial port.

## [FilterRegister](#)

Represents an operator that filters register-specific messages reported by the [AllenNeuralDynamics.WhiteRabbit](#) device.

## [Format](#)

Represents an operator which formats a sequence of values as specific WhiteRabbit register messages.

## [GetMetadata](#)

Represents an operator that returns the contents of the metadata file describing the [AllenNeuralDynamics.WhiteRabbit](#) device registers.

## [GroupByRegister](#)

Represents an operator that groups the sequence of [AllenNeuralDynamics.WhiteRabbit](#)" messages by register type.

## [Parse](#)

Represents an operator which filters and selects specific messages reported by the WhiteRabbit device.

#### [TimestampedAuxPortBaudRate](#)

Provides methods for manipulating timestamped messages from the AuxPortBaudRate register.

#### [TimestampedAuxPortMode](#)

Provides methods for manipulating timestamped messages from the AuxPortMode register.

#### [TimestampedConnectedDevices](#)

Provides methods for manipulating timestamped messages from the ConnectedDevices register.

#### [TimestampedCounter](#)

Provides methods for manipulating timestamped messages from the Counter register.

#### [TimestampedCounterFrequencyHz](#)

Provides methods for manipulating timestamped messages from the CounterFrequencyHz register.

## Enums

#### [AuxPortModeConfig](#)

Auxiliary port available configuration

#### [ClockOutChannels](#)

Clock output channels

# Class AsyncDevice

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents an asynchronous API to configure and interface with WhiteRabbit devices.

```
public class AsyncDevice : AsyncDevice, IDisposable
```

## Inheritance

[object](#) ← [AsyncDevice](#) ← AsyncDevice

## Implements

[IDisposable](#)

## Inherited Members

[AsyncDevice.ReadWhoAmIAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadHardwareVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadAssemblyVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadCoreVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadFirmwareVersionAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadTimestampSecondsAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadTimestampMicrosecondsAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadOperationControlAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadResetDeviceAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadDeviceNameAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadSerialNumberAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadClockConfigurationAsync\(CancellationToken\)](#) ,  
[AsyncDevice.ReadByteAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadByteArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSByteAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSByteArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt16Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt16ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt16Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt16ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt32Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt32ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt32Async\(int, CancellationToken\)](#) ,

[AsyncDevice.ReadInt32ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt64Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadUInt64ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt64Async\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadInt64ArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSingleAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.ReadSingleArrayAsync\(int, CancellationToken\)](#) ,  
[AsyncDevice.WriteTimestampSecondsAsync\(uint, CancellationToken\)](#) ,  
[AsyncDevice.WriteResetDeviceAsync\(ResetFlags, CancellationToken\)](#) ,  
[AsyncDevice.WriteDeviceNameAsync\(string, CancellationToken\)](#) ,  
[AsyncDevice.WriteClockConfigurationAsync\(ClockConfigurationFlags, CancellationToken\)](#) ,  
[AsyncDevice.WriteByteAsync\(int, byte, CancellationToken\)](#) ,  
[AsyncDevice.WriteByteAsync\(int, byte\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteSByteAsync\(int, sbyte, CancellationToken\)](#) ,  
[AsyncDevice.WriteSByteAsync\(int, sbyte\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteUInt16Async\(int, ushort, CancellationToken\)](#) ,  
[AsyncDevice.WriteUInt16Async\(int, ushort\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine16Async\(int, short, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine16Async\(int, short\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, uint, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, uint\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, int, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine32Async\(int, int\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, ulong, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, ulong\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, long, CancellationToken\)](#) ,  
[AsyncDevice.WriteLine64Async\(int, long\[\], CancellationToken\)](#) ,  
[AsyncDevice.WriteSingleAsync\(int, float, CancellationToken\)](#) ,  
[AsyncDevice.WriteSingleAsync\(int, float\[\], CancellationToken\)](#) ,  
[AsyncDevice.CommandAsync\(HarpMessage, CancellationToken\)](#) , [AsyncDevice.Dispose\(\)](#) ,  
[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Methods

### ReadAuxPortBaudRateAsync(CancellationToken)

Asynchronously reads the contents of the AuxPortBaudRate register.

```
public Task<uint> ReadAuxPortBaudRateAsync(CancellationToken cancellationToken = default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[uint](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadAuxPortModeAsync(CancellationToken)

Asynchronously reads the contents of the AuxPortMode register.

```
public Task<AuxPortModeConfig> ReadAuxPortModeAsync(CancellationToken cancellationToken  
= default)
```

## Parameters

cancellationToken [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[AuxPortModeConfig](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadConnectedDevicesAsync(CancellationToken)

Asynchronously reads the contents of the ConnectedDevices register.

```
public Task<ClockOutChannels> ReadConnectedDevicesAsync(CancellationToken cancellationToken = default)
```

## Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[ClockOutChannels](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadCounterAsync(CancellationToken)

Asynchronously reads the contents of the Counter register.

```
public Task<uint> ReadCounterAsync(CancellationToken cancellationToken = default)
```

## Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[uint](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadCounterFrequencyHzAsync(CancellationToken)

Asynchronously reads the contents of the CounterFrequencyHz register.

```
public Task<ushort> ReadCounterFrequencyHzAsync(CancellationToken cancellationToken  
= default)
```

## Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#) <[ushort](#)>

A task that represents the asynchronous read operation. The [Result](#) property contains the register payload.

## ReadTimestampedAuxPortBaudRateAsync(CancellationToken)

Asynchronously reads the timestamped contents of the AuxPortBaudRate register.

```
public Task<Timestamped<uint>> ReadTimestampedAuxPortBaudRateAsync(CancellationToken  
cancellationToken = default)
```

## Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#) <[Timestamped](#) <[uint](#)>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedAuxPortModeAsync(CancellationToken)

Asynchronously reads the timestamped contents of the AuxPortMode register.

```
public Task<Timestamped<AuxPortModeConfig>>
ReadTimestampedAuxPortModeAsync(CancellationToken cancellationToken = default)
```

## Parameters

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<[AuxPortModeConfig](#)>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedConnectedDevicesAsync(CancellationToken)

Asynchronously reads the timestamped contents of the ConnectedDevices register.

```
public Task<Timestamped<ClockOutChannels>>
ReadTimestampedConnectedDevicesAsync(CancellationToken cancellationToken = default)
```

## Parameters

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<[ClockOutChannels](#)>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedCounterAsync(CancellationToken)

Asynchronously reads the timestamped contents of the Counter register.

```
public Task<Timestamped<uint>> ReadTimestampedCounterAsync(CancellationToken  
cancellationToken = default)
```

## Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<uint>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## ReadTimestampedCounterFrequencyHzAsync(CancellationToken)

Asynchronously reads the timestamped contents of the CounterFrequencyHz register.

```
public Task<Timestamped<ushort>> ReadTimestampedCounterFrequencyHzAsync(CancellationToken  
cancellationToken = default)
```

## Parameters

`cancellationToken` [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)<[Timestamped](#)<ushort>>

A task that represents the asynchronous read operation. The [Result](#) property contains the timestamped register payload.

## WriteAuxPortBaudRateAsync(uint, CancellationToken)

Asynchronously writes a value to the AuxPortBaudRate register.

```
public Task WriteAuxPortBaudRateAsync(uint value, CancellationToken cancellationToken  
= default)
```

## Parameters

**value** [uint](#)

The value to be stored in the register.

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

[Task](#)

The task object representing the asynchronous write operation.

## WriteAuxPortModeAsync(AuxPortModeConfig, CancellationToken)

Asynchronously writes a value to the AuxPortMode register.

```
public Task WriteAuxPortModeAsync(AuxPortModeConfig value, CancellationToken  
cancellationToken = default)
```

## Parameters

**value** [AuxPortModeConfig](#)

The value to be stored in the register.

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

## Returns

## [Task](#)

The task object representing the asynchronous write operation.

## WriteCounterAsync(uint, CancellationToken)

Asynchronously writes a value to the Counter register.

```
public Task WriteCounterAsync(uint value, CancellationToken cancellationToken = default)
```

Parameters

**value** [uint](#)

The value to be stored in the register.

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

Returns

## [Task](#)

The task object representing the asynchronous write operation.

## WriteCounterFrequencyHzAsync(ushort, CancellationToken)

Asynchronously writes a value to the CounterFrequencyHz register.

```
public Task WriteCounterFrequencyHzAsync(ushort value, CancellationToken cancellationToken = default)
```

Parameters

**value** [ushort](#)

The value to be stored in the register.

**cancellationToken** [CancellationToken](#)

A [CancellationToken](#) which can be used to cancel the operation.

Returns

[Task](#)

The task object representing the asynchronous write operation.

# Class AuxPortBaudRate

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents a register that the baud rate, in bps, of the auxiliary port when in HarpClock mode.

```
public class AuxPortBaudRate
```

## Inheritance

[object](#) ← AuxPortBaudRate

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [AuxPortBaudRate](#) register. This field is constant.

```
public const int Address = 36
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [AuxPortBaudRate](#) register. This field is constant.

```
public const int RegisterLength = 1
```

### Field Value

## .RegisterType

Represents the payload type of the [AuxPortBaudRate](#) register. This field is constant.

```
public const PayloadType RegisterType = U32
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, uint)

Returns a Harp message for the [AuxPortBaudRate](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, uint value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [uint](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [AuxPortBaudRate](#) register with the specified message type and payload.

## FromPayload(double, MessageType, uint)

Returns a timestamped Harp message for the [AuxPortBaudRate](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType, uint value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [uint](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [AuxPortBaudRate](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [AuxPortBaudRate](#) register messages.

```
public static uint GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[uint](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [AuxPortBaudRate](#) register messages.

```
public static Timestamped<uint> GetTimestampedPayload(HarpMessage message)
```

### Parameters

[message](#) [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <[uint](#)>

A value representing the timestamped message payload.

# Class AuxPortMode

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents a register that the function of the auxiliary port.

```
public class AuxPortMode
```

## Inheritance

[object](#) ← AuxPortMode

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [AuxPortMode](#) register. This field is constant.

```
public const int Address = 35
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [AuxPortMode](#) register. This field is constant.

```
public const int RegisterLength = 1
```

### Field Value

## .RegisterType

Represents the payload type of the [AuxPortMode](#) register. This field is constant.

```
public const PayloadType RegisterType = U8
```

### Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, AuxPortModeConfig)

Returns a Harp message for the [AuxPortMode](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, AuxPortModeConfig value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [AuxPortModeConfig](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [AuxPortMode](#) register with the specified message type and payload.

### FromPayload(double, MessageType, AuxPortModeConfig)

Returns a timestamped Harp message for the [AuxPortMode](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
AuxPortModeConfig value)
```

Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [AuxPortModeConfig](#)

The value to be stored in the message payload.

Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [AuxPortMode](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [AuxPortMode](#) register messages.

```
public static AuxPortModeConfig GetPayload(HarpMessage message)
```

Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

Returns

[AuxPortModeConfig](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [AuxPortMode](#) register messages.

```
public static Timestamped<AuxPortModeConfig> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <[AuxPortModeConfig](#)>

A value representing the timestamped message payload.

# Enum AuxPortModeConfig

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Auxiliary port available configuration

```
public enum AuxPortModeConfig : byte
```

## Fields

Disabled = 0

HarpClock = 1

PPS = 2

# Enum ClockOutChannels

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Clock output channels

```
[Flags]
public enum ClockOutChannels : ushort
```

## Fields

Channel0 = 1

Channel1 = 2

Channel10 = 1024

Channel11 = 2048

Channel12 = 4096

Channel13 = 8192

Channel14 = 16384

Channel15 = 32768

Channel2 = 4

Channel3 = 8

Channel4 = 16

Channel5 = 32

Channel6 = 64

Channel7 = 128

Channel8 = 256

Channel9 = 512

None = 0

# Class ConnectedDevices

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents a register that the currently connected output channels. An event will be generated when any of the channels are connected or disconnected.

```
public class ConnectedDevices
```

## Inheritance

[object](#) ← ConnectedDevices

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [ConnectedDevices](#) register. This field is constant.

```
public const int Address = 32
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [ConnectedDevices](#) register. This field is constant.

```
public const int RegisterLength = 1
```

## Field Value

[int↗](#)

## RegisterType

Represents the payload type of the [ConnectedDevices](#) register. This field is constant.

```
public const PayloadType RegisterType = U16
```

## Field Value

[PayloadType↗](#)

## Methods

### FromPayload(MessageType, ClockOutChannels)

Returns a Harp message for the [ConnectedDevices](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, ClockOutChannels value)
```

#### Parameters

**messageType** [MessageType↗](#)

The type of the Harp message.

**value** [ClockOutChannels](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage↗](#)

A [HarpMessage↗](#) object for the [ConnectedDevices](#) register with the specified message type and payload.

## FromPayload(double, MessageType, ClockOutChannels)

Returns a timestamped Harp message for the [ConnectedDevices](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
ClockOutChannels value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [ClockOutChannels](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [ConnectedDevices](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [ConnectedDevices](#) register messages.

```
public static ClockOutChannels GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## [ClockOutChannels](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [ConnectedDevices](#) register messages.

```
public static Timestamped<ClockOutChannels> GetTimestampedPayload(HarpMessage message)
```

### Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <[ClockOutChannels](#)>

A value representing the timestamped message payload.

# Class Counter

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents a register that the counter value. This value is incremented at the frequency specified by CounterFrequencyHz. Write to force a counter value.

```
public class Counter
```

## Inheritance

[object](#) ← Counter

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Fields

## Address

Represents the address of the [Counter](#) register. This field is constant.

```
public const int Address = 33
```

## Field Value

[int](#)

## RegisterLength

Represents the length of the [Counter](#) register. This field is constant.

```
public const int RegisterLength = 1
```

## Field Value

[int↗](#)

## RegisterType

Represents the payload type of the [Counter](#) register. This field is constant.

```
public const PayloadType RegisterType = U32
```

## Field Value

[PayloadType↗](#)

## Methods

### FromPayload(MessageType, uint)

Returns a Harp message for the [Counter](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, uint value)
```

#### Parameters

**messageType** [MessageType↗](#)

The type of the Harp message.

**value** [uint↗](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage↗](#)

A [HarpMessage↗](#) object for the [Counter](#) register with the specified message type and payload.

## FromPayload(double, MessageType, uint)

Returns a timestamped Harp message for the [Counter](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType, uint value)
```

Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [uint](#)

The value to be stored in the message payload.

Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [Counter](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [Counter](#) register messages.

```
public static uint GetPayload(HarpMessage message)
```

Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

Returns

[uint](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [Counter](#) register messages.

```
public static Timestamped<uint> GetTimestampedPayload(HarpMessage message)
```

### Parameters

[message](#) [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) <[uint](#)>

A value representing the timestamped message payload.

# Class CounterFrequencyHz

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents a register that the frequency at which the counter is incremented. A value of 0 disables the counter.

```
public class CounterFrequencyHz
```

## Inheritance

[object](#) ← CounterFrequencyHz

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [CounterFrequencyHz](#) register. This field is constant.

```
public const int Address = 34
```

### Field Value

[int](#)

### RegisterLength

Represents the length of the [CounterFrequencyHz](#) register. This field is constant.

```
public const int RegisterLength = 1
```

## Field Value

[int](#)

## RegisterType

Represents the payload type of the [CounterFrequencyHz](#) register. This field is constant.

```
public const PayloadType RegisterType = U16
```

## Field Value

[PayloadType](#)

## Methods

### FromPayload(MessageType, ushort)

Returns a Harp message for the [CounterFrequencyHz](#) register.

```
public static HarpMessage FromPayload(MessageType messageType, ushort value)
```

#### Parameters

**messageType** [MessageType](#)

The type of the Harp message.

**value** [ushort](#)

The value to be stored in the message payload.

#### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [CounterFrequencyHz](#) register with the specified message type and payload.

## FromPayload(double, MessageType, ushort)

Returns a timestamped Harp message for the [CounterFrequencyHz](#) register.

```
public static HarpMessage FromPayload(double timestamp, MessageType messageType,  
ushort value)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

The type of the Harp message.

**value** [ushort](#)

The value to be stored in the message payload.

### Returns

[HarpMessage](#)

A [HarpMessage](#) object for the [CounterFrequencyHz](#) register with the specified message type, timestamp, and payload.

## GetPayload(HarpMessage)

Returns the payload data for [CounterFrequencyHz](#) register messages.

```
public static ushort GetPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

## [ushort](#)

A value representing the message payload.

## GetTimestampedPayload(HarpMessage)

Returns the timestamped payload data for [CounterFrequencyHz](#) register messages.

```
public static Timestamped<ushort> GetTimestampedPayload(HarpMessage message)
```

### Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

### Returns

[Timestamped](#) < ushort >

A value representing the timestamped message payload.

# Class CreateAuxPortBaudRatePayload

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents an operator that creates a message payload that the baud rate, in bps, of the auxiliary port when in HarpClock mode.

```
public class CreateAuxPortBaudRatePayload
```

## Inheritance

[object](#) ← CreateAuxPortBaudRatePayload

## Derived

[CreateTimestampedAuxPortBaudRatePayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## AuxPortBaudRate

Gets or sets the value that the baud rate, in bps, of the auxiliary port when in HarpClock mode.

```
[Range(40, 1000000)]  
public uint AuxPortBaudRate { get; set; }
```

## Property Value

[uint](#)

# Methods

## GetMessage(MessageType)

Creates a message that the baud rate, in bps, of the auxiliary port when in HarpClock mode.

```
public HarpMessage GetMessage(MessageType messageType)
```

Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

Returns

[HarpMessage](#)

A new message for the AuxPortBaudRate register.

## GetPayload()

Creates a message payload for the AuxPortBaudRate register.

```
public uint GetPayload()
```

Returns

[uint](#)

The created message payload value.

# Class CreateAuxPortModePayload

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents an operator that creates a message payload that the function of the auxiliary port.

```
public class CreateAuxPortModePayload
```

## Inheritance

[object](#) ← CreateAuxPortModePayload

## Derived

[CreateTimestampedAuxPortModePayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Properties

### AuxPortMode

Gets or sets the value that the function of the auxiliary port.

```
public AuxPortModeConfig AuxPortMode { get; set; }
```

## Property Value

[AuxPortModeConfig](#)

## Methods

### GetMessage(MessageType)

Creates a message that the function of the auxiliary port.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the AuxPortMode register.

## GetPayload()

Creates a message payload for the AuxPortMode register.

```
public AuxPortModeConfig GetPayload()
```

## Returns

[AuxPortModeConfig](#)

The created message payload value.

# Class CreateConnectedDevicesPayload

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents an operator that creates a message payload that the currently connected output channels. An event will be generated when any of the channels are connected or disconnected.

```
public class CreateConnectedDevicesPayload
```

## Inheritance

[object](#) ← CreateConnectedDevicesPayload

## Derived

[CreateTimestampedConnectedDevicesPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## ConnectedDevices

Gets or sets the value that the currently connected output channels. An event will be generated when any of the channels are connected or disconnected.

```
public ClockOutChannels ConnectedDevices { get; set; }
```

## Property Value

[ClockOutChannels](#)

# Methods

## GetMessage(MessageType)

Creates a message that the currently connected output channels. An event will be generated when any of the channels are connected or disconnected.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the ConnectedDevices register.

## GetPayload()

Creates a message payload for the ConnectedDevices register.

```
public ClockOutChannels GetPayload()
```

## Returns

[ClockOutChannels](#)

The created message payload value.

# Class CreateCounterFrequencyHzPayload

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents an operator that creates a message payload that the frequency at which the counter is incremented. A value of 0 disables the counter.

```
public class CreateCounterFrequencyHzPayload
```

## Inheritance

[object](#) ← CreateCounterFrequencyHzPayload

## Derived

[CreateTimestampedCounterFrequencyHzPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## CounterFrequencyHz

Gets or sets the value that the frequency at which the counter is incremented. A value of 0 disables the counter.

```
[Range(0, 500)]  
public ushort CounterFrequencyHz { get; set; }
```

## Property Value

[ushort](#)

# Methods

## GetMessage(MessageType)

Creates a message that the frequency at which the counter is incremented. A value of 0 disables the counter.

```
public HarpMessage GetMessage(MessageType messageType)
```

### Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

### Returns

[HarpMessage](#)

A new message for the CounterFrequencyHz register.

## GetPayload()

Creates a message payload for the CounterFrequencyHz register.

```
public ushort GetPayload()
```

### Returns

[ushort](#)

The created message payload value.

# Class CreateCounterPayload

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents an operator that creates a message payload that the counter value. This value is incremented at the frequency specified by CounterFrequencyHz. Write to force a counter value.

```
public class CreateCounterPayload
```

## Inheritance

[object](#) ← CreateCounterPayload

## Derived

[CreateTimestampedCounterPayload](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Counter

Gets or sets the value that the counter value. This value is incremented at the frequency specified by CounterFrequencyHz. Write to force a counter value.

```
public uint Counter { get; set; }
```

## Property Value

[uint](#)

# Methods

## GetMessage(MessageType)

Creates a message that the counter value. This value is incremented at the frequency specified by CounterFrequencyHz. Write to force a counter value.

```
public HarpMessage GetMessage(MessageType messageType)
```

## Parameters

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new message for the Counter register.

## GetPayload()

Creates a message payload for the Counter register.

```
public uint GetPayload()
```

## Returns

[uint](#)

The created message payload value.

# Class CreateMessage

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents an operator which creates standard message payloads for the WhiteRabbit device.

```
public class CreateMessage : CreateMessageBuilder, IExpressionBuilder,  
ICustomTypeDescriptor, INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [CreateMessageBuilder](#) ← CreateMessage

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[CreateMessageBuilder.Build\(IEnumerable<Expression>\)](#), [CreateMessageBuilder.ArgumentRange](#),  
[CreateMessageBuilder.MessageType](#), [CreateMessageBuilder.Payload](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

## Constructors

### CreateMessage()

Initializes a new instance of the [CreateMessage](#) class.

```
public CreateMessage()
```

## See Also

[CreateConnectedDevicesPayload](#)

[CreateCounterPayload](#)

[CreateCounterFrequencyHzPayload](#)

[CreateAuxPortModePayload](#)

[CreateAuxPortBaudRatePayload](#)

# Class

## CreateTimestampedAuxPortBaudRatePayload

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents an operator that creates a timestamped message payload that the baud rate, in bps, of the auxiliary port when in HarpClock mode.

```
public class CreateTimestampedAuxPortBaudRatePayload : CreateAuxPortBaudRatePayload
```

### Inheritance

[object](#) ← [CreateAuxPortBaudRatePayload](#) ← CreateTimestampedAuxPortBaudRatePayload

### Inherited Members

[CreateAuxPortBaudRatePayload.AuxPortBaudRate](#) , [CreateAuxPortBaudRatePayload.GetPayload\(\)](#) ,  
[CreateAuxPortBaudRatePayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that the baud rate, in bps, of the auxiliary port when in HarpClock mode.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

[HarpMessage](#)

A new timestamped message for the AuxPortBaudRate register.

# Class CreateTimestampedAuxPortModePayload

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents an operator that creates a timestamped message payload that the function of the auxiliary port.

```
public class CreateTimestampedAuxPortModePayload : CreateAuxPortModePayload
```

## Inheritance

[object](#) ← [CreateAuxPortModePayload](#) ← CreateTimestampedAuxPortModePayload

## Inherited Members

[CreateAuxPortModePayload.AuxPortMode](#) , [CreateAuxPortModePayload.GetPayload\(\)](#) ,  
[CreateAuxPortModePayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that the function of the auxiliary port.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

#### Returns

## [HarpMessage](#)

A new timestamped message for the AuxPortMode register.

# Class

# CreateTimestampedConnectedDevicesPayload

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents an operator that creates a timestamped message payload that the currently connected output channels. An event will be generated when any of the channels are connected or disconnected.

```
public class CreateTimestampedConnectedDevicesPayload : CreateConnectedDevicesPayload
```

## Inheritance

[object](#) ← [CreateConnectedDevicesPayload](#) ← CreateTimestampedConnectedDevicesPayload

## Inherited Members

[CreateConnectedDevicesPayload.ConnectedDevices](#) , [CreateConnectedDevicesPayload.GetPayload\(\)](#) ,  
[CreateConnectedDevicesPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

# Methods

## GetMessage(double, MessageType)

Creates a timestamped message that the currently connected output channels. An event will be generated when any of the channels are connected or disconnected.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

### [HarpMessage](#)

A new timestamped message for the ConnectedDevices register.

# Class

# CreateTimestampedCounterFrequencyHzPayload

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents an operator that creates a timestamped message payload that the frequency at which the counter is incremented. A value of 0 disables the counter.

```
public class CreateTimestampedCounterFrequencyHzPayload : CreateCounterFrequencyHzPayload
```

## Inheritance

[object](#) ← [CreateCounterFrequencyHzPayload](#) ← CreateTimestampedCounterFrequencyHzPayload

## Inherited Members

[CreateCounterFrequencyHzPayload.CounterFrequencyHz](#) ,  
[CreateCounterFrequencyHzPayload.GetPayload\(\)](#) ,  
[CreateCounterFrequencyHzPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

# Methods

## GetMessage(double, MessageType)

Creates a timestamped message that the frequency at which the counter is incremented. A value of 0 disables the counter.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

## Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

Returns

[HarpMessage](#)

A new timestamped message for the CounterFrequencyHz register.

# Class CreateTimestampedCounterPayload

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents an operator that creates a timestamped message payload that the counter value. This value is incremented at the frequency specified by CounterFrequencyHz. Write to force a counter value.

```
public class CreateTimestampedCounterPayload : CreateCounterPayload
```

## Inheritance

[object](#) ← [CreateCounterPayload](#) ← CreateTimestampedCounterPayload

## Inherited Members

[CreateCounterPayload.Counter](#) , [CreateCounterPayload.GetPayload\(\)](#) ,  
[CreateCounterPayload.GetMessage\(MessageType\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Methods

### GetMessage(double, MessageType)

Creates a timestamped message that the counter value. This value is incremented at the frequency specified by CounterFrequencyHz. Write to force a counter value.

```
public HarpMessage GetMessage(double timestamp, MessageType messageType)
```

#### Parameters

**timestamp** [double](#)

The timestamp of the message payload, in seconds.

**messageType** [MessageType](#)

Specifies the type of the created message.

## Returns

### [HarpMessage](#) ↴

A new timestamped message for the Counter register.

# Class Device

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents an observable source of messages from the Harp device connected at the specified serial port.

```
[Combinator(MethodName = "Generate")]
[WorkflowElementCategory(ElementCategory.Source)]
public class Device : Device, INamedElement
```

## Inheritance

[object](#) ← [Source](#) < [HarpMessage](#) > ← [Device](#) ← [Device](#)

## Implements

[INamedElement](#)

## Inherited Members

[Device.Generate\(\)](#) , [Device.Generate\(IObservable<HarpMessage>\)](#) , [Device.OperationMode](#) ,  
[Device.OperationLed](#) , [Device.DumpRegisters](#) , [Device.VisualIndicators](#) , [Device.Heartbeat](#) ,  
[Device.IgnoreErrors](#) , [Device.PortName](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

# Constructors

## Device()

Initializes a new instance of the [Device](#) class.

```
public Device()
```

# Fields

## Metadata

Gets the contents of the metadata file describing the [AllenNeuralDynamics.WhiteRabbit](#) device registers.

```
public static readonly string Metadata
```

Field Value

[string](#)

## WhoAmI

Represents the unique identity class of the [AllenNeuralDynamics.WhiteRabbit](#) device. This field is constant.

```
public const int WhoAmI = 1404
```

Field Value

[int](#)

## Properties

### RegisterMap

Gets a read-only mapping from address to register type.

```
public static IReadOnlyDictionary<int, Type> RegisterMap { get; }
```

Property Value

[IReadOnlyDictionary](#)<[int](#), [Type](#)>

## Methods

### CreateAsync(string)

Initializes a new instance of the asynchronous API to configure and interface with WhiteRabbit devices on the specified serial port.

```
public static Task<AsyncDevice> CreateAsync(string portName)
```

## Parameters

**portName** [string](#)

The name of the serial port used to communicate with the Harp device.

## Returns

[Task](#) <[AsyncDevice](#)>

A task that represents the asynchronous initialization operation. The value of the [Result](#) parameter contains a new instance of the [AsyncDevice](#) class.

# Class FilterRegister

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents an operator that filters register-specific messages reported by the [AllenNeuralDynamics.WhiteRabbit](#) device.

```
public class FilterRegister : FilterRegisterBuilder, IExpressionBuilder,  
ICustomTypeDescriptor, INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [RegisterCombinatorBuilder](#) ←  
[FilterRegisterBuilder](#) ← FilterRegister

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[FilterRegisterBuilder.FilterType](#), [FilterRegisterBuilder.Register](#),  
[RegisterCombinatorBuilder.Build\(IEnumerable<Expression>\)](#),  
[RegisterCombinatorBuilder.ArgumentRange](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

## Constructors

### FilterRegister()

Initializes a new instance of the [FilterRegister](#) class.

```
public FilterRegister()
```

## See Also

[ConnectedDevices](#)

[Counter](#)

[CounterFrequencyHz](#)

[AuxPortMode](#)

[AuxPortBaudRate](#)

# Class Format

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents an operator which formats a sequence of values as specific WhiteRabbit register messages.

```
public class Format : FormatBuilder, IExpressionBuilder, ICustomTypeDescriptor,  
INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [FormatBuilder](#) ← Format

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[FormatBuilder.Build\(IEnumerable<Expression>\)](#), [FormatBuilder.ArgumentRange](#),  
[FormatBuilder.MessageType](#), [FormatBuilder.Register](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

## Constructors

### Format()

Initializes a new instance of the [Format](#) class.

```
public Format()
```

## See Also

[ConnectedDevices](#)

[Counter](#)

[CounterFrequencyHz](#)

[AuxPortMode](#)

[AuxPortBaudRate](#)

# Class GetMetadata

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents an operator that returns the contents of the metadata file describing the [AllenNeuralDynamics.WhiteRabbit](#) device registers.

```
public class GetMetadata : Source<string>
```

## Inheritance

[object](#) ← [Source](#)<[string](#)> ← GetMetadata

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Methods

### Generate()

Returns an observable sequence with the contents of the metadata file describing the [AllenNeuralDynamics.WhiteRabbit](#) device registers.

```
public override IObservable<string> Generate()
```

## Returns

[IObservable](#)<[string](#)>

A sequence with a single [string](#) object representing the contents of the metadata file.

# Class GroupByRegister

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents an operator that groups the sequence of [AllenNeuralDynamics.WhiteRabbit](#)" messages by register type.

```
public class GroupByRegister : Combinator<HarpMessage, IGroupedObservable<Type, HarpMessage>>
```

## Inheritance

```
object ↗ ← Combinator ↗ <HarpMessage ↗, IGroupedObservable ↗ <Type ↗, HarpMessage ↗>> ← GroupByRegister
```

## Inherited Members

```
Combinator<HarpMessage, IGroupedObservable<Type, HarpMessage>>.Process(I Observable<HarpMessage>) ↗ ,  
object.ToString() ↗ , object.Equals(object) ↗ , object.Equals(object, object) ↗ ,  
object.ReferenceEquals(object, object) ↗ , object.GetHashCode() ↗ , object.GetType() ↗ ,  
object.MemberwiseClone() ↗
```

## Methods

### Process(I Observable<HarpMessage>)

Groups an observable sequence of [AllenNeuralDynamics.WhiteRabbit](#) messages by register type.

```
public override IObservable<IGroupedObservable<Type, HarpMessage>>  
Process(IObservable<HarpMessage> source)
```

## Parameters

source [IObservable](#)<HarpMessage>

The sequence of Harp device messages.

## Returns

[IObservable](#) < [IGroupedObservable](#) < [Type](#), [HarpMessage](#) >>

A sequence of observable groups, each of which corresponds to a unique [AllenNeuralDynamics.White Rabbit](#) register.

# Class Parse

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Represents an operator which filters and selects specific messages reported by the WhiteRabbit device.

```
public class Parse : ParseBuilder, IExpressionBuilder, ICustomTypeDescriptor, INamedElement
```

## Inheritance

[object](#) ← [ExpressionBuilder](#) ← [HarpCombinatorBuilder](#) ← [RegisterCombinatorBuilder](#) ← [ParseBuilder](#) ← Parse

## Implements

[IExpressionBuilder](#), [ICustomTypeDescriptor](#), [INamedElement](#)

## Inherited Members

[ParseBuilder.Register](#), [RegisterCombinatorBuilder.Build\(IEnumerable<Expression>\)](#),  
[RegisterCombinatorBuilder.ArgumentRange](#), [ExpressionBuilder.ToString\(\)](#),  
[ExpressionBuilder.Unwrap\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetWorkflowElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerElement\(ExpressionBuilder\)](#),  
[ExpressionBuilder.GetVisualizerMappings\(ExpressionBuilder\)](#),  
[ExpressionBuilder.FromWorkflowElement\(object, ElementCategory\)](#),  
[ExpressionBuilder.GetElementDisplayName\(Type\)](#),  
[ExpressionBuilder.GetElementDisplayName\(object\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

## Constructors

### Parse()

Initializes a new instance of the [Parse](#) class.

```
public Parse()
```

## See Also

[ConnectedDevices](#)

[Counter](#)

[CounterFrequencyHz](#)

[AuxPortMode](#)

[AuxPortBaudRate](#)

# Class TimestampedAuxPortBaudRate

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Provides methods for manipulating timestamped messages from the AuxPortBaudRate register.

```
public class TimestampedAuxPortBaudRate
```

## Inheritance

[object](#) ← TimestampedAuxPortBaudRate

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [AuxPortBaudRate](#) register. This field is constant.

```
public const int Address = 36
```

### Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [AuxPortBaudRate](#) register messages.

```
public static Timestamped<uint> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[uint](#)>

A value representing the timestamped message payload.

## See Also

[AuxPortBaudRate](#)

# Class TimestampedAuxPortMode

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Provides methods for manipulating timestamped messages from the AuxPortMode register.

```
public class TimestampedAuxPortMode
```

## Inheritance

[object](#) ← TimestampedAuxPortMode

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [AuxPortMode](#) register. This field is constant.

```
public const int Address = 35
```

### Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [AuxPortMode](#) register messages.

```
public static Timestamped<AuxPortModeConfig> GetPayload(HarpMessage message)
```

## Parameters

message [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[AuxPortModeConfig](#)>

A value representing the timestamped message payload.

## See Also

[AuxPortMode](#)

# Class TimestampedConnectedDevices

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Provides methods for manipulating timestamped messages from the ConnectedDevices register.

```
public class TimestampedConnectedDevices
```

## Inheritance

[object](#) ← TimestampedConnectedDevices

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [ConnectedDevices](#) register. This field is constant.

```
public const int Address = 32
```

## Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [ConnectedDevices](#) register messages.

```
public static Timestamped<ClockOutChannels> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[ClockOutChannels](#)>

A value representing the timestamped message payload.

## See Also

[ConnectedDevices](#)

# Class TimestampedCounter

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Provides methods for manipulating timestamped messages from the Counter register.

```
public class TimestampedCounter
```

## Inheritance

[object](#) ← TimestampedCounter

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [Counter](#) register. This field is constant.

```
public const int Address = 33
```

## Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [Counter](#) register messages.

```
public static Timestamped<uint> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[uint](#)>

A value representing the timestamped message payload.

## See Also

[Counter](#)

# Class TimestampedCounterFrequencyHz

Namespace: [AllenNeuralDynamics.WhiteRabbit](#)

Assembly: AllenNeuralDynamics.WhiteRabbit.dll

Provides methods for manipulating timestamped messages from the CounterFrequencyHz register.

```
public class TimestampedCounterFrequencyHz
```

## Inheritance

[object](#) ← TimestampedCounterFrequencyHz

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Fields

### Address

Represents the address of the [CounterFrequencyHz](#) register. This field is constant.

```
public const int Address = 34
```

## Field Value

[int](#)

## Methods

### GetPayload(HarpMessage)

Returns timestamped payload data for [CounterFrequencyHz](#) register messages.

```
public static Timestamped<ushort> GetPayload(HarpMessage message)
```

## Parameters

**message** [HarpMessage](#)

A [HarpMessage](#) object representing the register message.

## Returns

[Timestamped](#) <[ushort](#)>

A value representing the timestamped message payload.

## See Also

[CounterFrequencyHz](#)

# Namespace AllenNeuralDynamics.Zaber

## Classes

[AccelerationUnitsConverter](#)

[CreateZaberDevice](#)

Represents an operator that establishes a connection with a [ZaberDevice](#).

[GenericCommand](#)

Represents an operator that sends a command to a [ZaberDevice](#) expecting a single Zaber.Motion.Ascii.Response.

[GenericCommandMultiResponse](#)

Represents an operator that sends a command to a [ZaberDevice](#) expecting multiple Zaber.Motion.Ascii.Response.

[GenericCommandNoResponse](#)

Represents an operator sends a command to a [ZaberDevice](#) without expecting a reply.

[GetPosition](#)

Represents an operator queries the current position of a [ZaberDevice](#) axis.

[GetSetting](#)

Represents an operator that sends reads a setting from a given axis..

[Home](#)

Represents an operator that instructs a [ZaberDevice](#) to perform a homing routine.

[IsBusy](#)

Represents an operator queries the current state of a [ZaberDevice](#) axis.

[MoveAbsolute](#)

Represents an operator that instructs a [ZaberDevice](#) move to the target absolute position.

[MoveRelative](#)

Represents an operator that instructs a [ZaberDevice](#) move to a relative position.

[MoveVelocity](#)

Represents an operator that instructs a [ZaberDevice](#) move with a given velocity value.

[Park](#)

Represents an operator that instructs a [ZaberDevice](#) to park, or unpark, a selected axis or device.

[PositionUnitsConverter](#)

## [SetSetting](#)

Represents an operator that sets a setting for a specified axis.

## [Stop](#)

Represents an operator that instructs a [ZaberDevice](#) to stop any movement on a selected axis.

## [VelocityUnitsConverter](#)

## [WaitUntilIdle](#)

Represents an operator that queries and waits to return until a [ZaberDevice](#) goes idle.

## [ZaberDevice](#)

Represents an Zaber manipulator communicating with the host computer using the Zaber.Motion SDK.

## [ZaberDeviceConfiguration](#)

Represents configuration settings used to initialize a connection with a [ZaberDevice](#)

# Class AccelerationUnitsConverter

Namespace: [AllenNeuralDynamics.Zaber](#)

Assembly: AllenNeuralDynamics.Zaber.dll

```
public class AccelerationUnitsConverter : EnumConverter
```

## Inheritance

[object](#) ← [TypeConverter](#) ← [EnumConverter](#) ← AccelerationUnitsConverter

## Inherited Members

[EnumConverter.CanConvertFrom\(ITypeDescriptorContext, Type\)](#) ,  
[EnumConverter.CanConvertTo\(ITypeDescriptorContext, Type\)](#) ,  
[EnumConverter.ConvertFrom\(ITypeDescriptorContext, CultureInfo, object\)](#) ,  
[EnumConverter.ConvertTo\(ITypeDescriptorContext, CultureInfo, object, Type\)](#) ,  
[EnumConverter.GetStandardValuesExclusive\(ITypeDescriptorContext\)](#) ,  
[EnumConverter.GetStandardValuesSupported\(ITypeDescriptorContext\)](#) ,  
[EnumConverter.IsValid\(ITypeDescriptorContext, object\)](#) , [EnumConverter.EnumType](#) ,  
[EnumConverter.Values](#) , [EnumConverter.Comparer](#) , [TypeConverter.CanConvertFrom\(Type\)](#) ,  
[TypeConverter.CanConvertTo\(Type\)](#) , [TypeConverter.ConvertFrom\(object\)](#) ,  
[TypeConverter.ConvertFromInvariantString\(string\)](#) ,  
[TypeConverter.ConvertFromInvariantString\(ITypeDescriptorContext, string\)](#) ,  
[TypeConverter.ConvertFromString\(string\)](#) ,  
[TypeConverter.ConvertFromString\(ITypeDescriptorContext, string\)](#) ,  
[TypeConverter.ConvertFromString\(ITypeDescriptorContext, CultureInfo, string\)](#) ,  
[TypeConverter.ConvertTo\(object, Type\)](#) , [TypeConverter.ConvertToInvariantString\(object\)](#) ,  
[TypeConverter.ConvertToInvariantString\(ITypeDescriptorContext, object\)](#) ,  
[TypeConverter.ConvertToString\(object\)](#) ,  
[TypeConverter.ConvertToString\(ITypeDescriptorContext, object\)](#) ,  
[TypeConverter.ConvertToString\(ITypeDescriptorContext, CultureInfo, object\)](#) ,  
[TypeConverter.CreateInstance\(IDictionary\)](#) ,  
[TypeConverter.CreateInstance\(ITypeDescriptorContext, IDictionary\)](#) ,  
[TypeConverter.GetConvertFromException\(object\)](#) ,  
[TypeConverter.GetConvertToException\(object, Type\)](#) , [TypeConverter.GetCreateInstanceSupported\(\)](#) ,  
[TypeConverter.GetCreateInstanceSupported\(ITypeDescriptorContext\)](#) ,  
[TypeConverter.GetProperties\(object\)](#) , [TypeConverter.GetProperties\(ITypeDescriptorContext, object\)](#) ,  
[TypeConverter.GetProperties\(ITypeDescriptorContext, object, Attribute\[\]\)](#) ,  
[TypeConverter.GetPropertiesSupported\(\)](#) ,  
[TypeConverter.GetPropertiesSupported\(ITypeDescriptorContext\)](#) ,

[TypeConverter.GetStandardValues\(\)](#) , [TypeConverter.GetStandardValuesExclusive\(\)](#) ,  
[TypeConverter.GetStandardValuesSupported\(\)](#) , [TypeConverter.IsValid\(object\)](#) ,  
[TypeConverter.SortProperties\(PropertyDescriptorCollection, string\[\]\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### AccelerationUnitsConverter()

```
public AccelerationUnitsConverter()
```

## Methods

### GetStandardValues(ITypeDescriptorContext)

Gets a collection of standard values for the data type this validator is designed for.

```
public override TypeConverter.StandardValuesCollection  
GetStandardValues(ITypeDescriptorContext context)
```

#### Parameters

context [ITypeDescriptorContext](#)

An [ITypeDescriptorContext](#) that provides a format context.

#### Returns

[TypeConverter](#).[StandardValuesCollection](#)

A [TypeConverter.StandardValuesCollection](#) that holds a standard set of valid values, or [null](#) if the data type does not support a standard set of values.

# Class CreateZaberDevice

Namespace: [AllenNeuralDynamics.Zaber](#)

Assembly: AllenNeuralDynamics.Zaber.dll

Represents an operator that establishes a connection with a [ZaberDevice](#).

```
public class CreateZaberDevice : Source<ZaberDevice>, INamedElement
```

## Inheritance

[object](#) ← [Source](#)<[ZaberDevice](#)> ← CreateZaberDevice

## Implements

[INamedElement](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Name

Gets or sets the optional alias to be used for the manipulator.

```
public string Name { get; set; }
```

## Property Value

[string](#)

## PortName

Gets or sets the COM port where the manipulator is connected to.

```
[TypeConverter(typeof(SerialPortNameConverter))]  
public string PortName { get; set; }
```

Property Value

[string](#)

## Methods

### Generate()

Generates an observable with a single [ZaberDevice](#) resource object.

```
public override IObservable<ZaberDevice> Generate()
```

Returns

[IObservable](#)<[ZaberDevice](#)>

A sequence of a single [ZaberDevice](#) resource object.

# Class GenericCommand

Namespace: [AllenNeuralDynamics.Zaber](#)

Assembly: AllenNeuralDynamics.Zaber.dll

Represents an operator that sends a command to a [ZaberDevice](#) expecting a single Zaber.Motion.Ascii. Response.

```
public class GenericCommand : Combinator<string, Response>
```

## Inheritance

[object](#) ← [Combinator](#)<[string](#), Response> ← GenericCommand

## Inherited Members

[Combinator<string, Response>.Process\(IObservable<string>\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

# Properties

## Axis

Gets or sets the axis of the manipulator to be controlled.

```
public int? Axis { get; set; }
```

## Property Value

[int](#)?

## Device

Gets or sets the device to be controlled. Defaults to 0.

```
public int? Device { get; set; }
```

## Property Value

[int](#)?

## PortName

Gets or sets the COM port or alias of the target [ZaberDevice](#)

```
[TypeConverter(typeof(PortNameConverter))]  
public string PortName { get; set; }
```

## Property Value

[string](#)

## Methods

### Process(I`Observable<string>`)

Sends a command to the device expecting a single reply.

```
public override Ibservable<Response> Process(Ibservable<string> source)
```

#### Parameters

source [I`bservable<string>`](#)

#### Returns

[I`bservable<Response>`](#)

On each event, it will emit the response of the device.

# Class GenericCommandMultiResponse

Namespace: [AllenNeuralDynamics.Zaber](#)

Assembly: AllenNeuralDynamics.Zaber.dll

Represents an operator that sends a command to a [ZaberDevice](#) expecting multiple Zaber.Motion.Ascii.Response.

```
public class GenericCommandMultiResponse : Combinator<string, Response[]>
```

## Inheritance

[object](#) ← [Combinator](#)<[string](#), Response[]> ← GenericCommandMultiResponse

## Inherited Members

[Combinator<string, Response\[\]>.Process\(IObservable<string>\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

# Properties

## Axis

Gets or sets the axis of the manipulator to be controlled.

```
public int? Axis { get; set; }
```

## Property Value

[int](#)?

## Device

Gets or sets the device to be controlled. Defaults to 0.

```
public int? Device { get; set; }
```

## Property Value

[int](#)?

## PortName

Gets or sets the COM port or alias of the target [ZaberDevice](#)

```
[TypeConverter(typeof(PortNameConverter))]  
public string PortName { get; set; }
```

## Property Value

[string](#)

## Methods

### Process(I`Observable<string>`)

Sends a command to the device expecting multiple replies.

```
public override Ibservable<Response[]> Process(Ibservable<string> source)
```

#### Parameters

source [I`bservable<string>`](#)

#### Returns

[I`bservable<Response\[\]>`](#)

On each event, it will emit the response of the device.

# Class GenericCommandNoResponse

Namespace: [AllenNeuralDynamics.Zaber](#)

Assembly: AllenNeuralDynamics.Zaber.dll

Represents an operator sends a command to a [ZaberDevice](#) without expecting a reply.

```
public class GenericCommandNoResponse : Sink<string>
```

## Inheritance

[object](#) ← [Combinator<string, string>](#) ← [Sink<string>](#) ← GenericCommandNoResponse

## Inherited Members

[Combinator<string, string>.Process\(IObservable<string>\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Properties

### Axis

Gets or sets the axis of the manipulator to be controlled.

```
public int? Axis { get; set; }
```

### Property Value

[int](#)?

### Device

Gets or sets the device to be controlled. Defaults to 0.

```
public int? Device { get; set; }
```

### Property Value

[int](#)?

## PortName

Gets or sets the COM port or alias of the target [ZaberDevice](#)

```
[TypeConverter(typeof(PortNameConverter))]  
public string PortName { get; set; }
```

### Property Value

[string](#)

## Methods

### Process(I`Observable<string>`)

Sends a command to the device without expecting a reply.

```
public override Ibservable<string> Process(Ibservable<string> source)
```

### Parameters

source [I`bservable<string>`](#)

### Returns

[I`bservable<string>`](#)

It will propagate the original sequence.

# Class GetPosition

Namespace: [AllenNeuralDynamics.Zaber](#)

Assembly: AllenNeuralDynamics.Zaber.dll

Represents an operator queries the current position of a [ZaberDevice](#) axis.

```
public class GetPosition : Source<double>
```

## Inheritance

[object](#) ← [Source](#)<[double](#)> ← GetPosition

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Axis

Gets or sets the axis of the manipulator to be controlled.

```
public int Axis { get; set; }
```

## Property Value

[int](#)

## Device

Gets or sets the device to be controlled. Defaults to 0.

```
public int Device { get; set; }
```

## Property Value

[int ↗](#)

## PortName

Gets or sets the COM port or alias of the target [ZaberDevice](#)

```
[TypeConverter(typeof(PortNameConverter))]  
public string PortName { get; set; }
```

### Property Value

[string ↗](#)

## Units

Gets or sets the Units of position the manipulator instruction is operating on.

```
[TypeConverter(typeof(PositionUnitsConverter))]  
public Units Units { get; set; }
```

### Property Value

Units

## Methods

### Generate()

Queries the current position of an axis from the manipulator.

```
public override IObservable<double> Generate()
```

### Returns

[IObservable ↗ <double ↗ >](#)

When subscribed, it will emit a sequence with a single value of the most recent axis position.

## Generate<TSource>(IObservable<TSource>)

Queries the current position of an axis from the manipulator on each event.

```
public IObservable<double> Generate<TSource>(IObservable<TSource> source)
```

Parameters

source [IObservable](#)<TSource>

Returns

[IObservable](#)<double>

On each event, it will emit a value with a single value of the most recent axis position.

Type Parameters

TSource

# Class GetSetting

Namespace: [AllenNeuralDynamics.Zaber](#)

Assembly: AllenNeuralDynamics.Zaber.dll

Represents an operator that sends reads a setting from a given axis..

```
public class GetSetting : Combinator<double>
```

## Inheritance

[object](#) ← [Combinator](#)<[double](#)> ← GetSetting

## Inherited Members

[Combinator<double>.Process<TSource>\(IObservable<TSource>\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

# Properties

## Axis

Gets or sets the axis of the manipulator to be controlled.

```
public int Axis { get; set; }
```

## Property Value

[int](#)

## Device

Gets or sets the device to be controlled. Defaults to 0.

```
public int? Device { get; set; }
```

## Property Value

[int](#)?

## PortName

Gets or sets the COM port or alias of the target [ZaberDevice](#)

```
[TypeConverter(typeof(PortNameConverter))]  
public string PortName { get; set; }
```

### Property Value

[string](#)

## Setting

Gets or sets the axis setting to be queried.

```
public string Setting { get; set; }
```

### Property Value

[string](#)

## Units

Gets or sets the Units of the transaction.

```
public Units Units { get; set; }
```

### Property Value

Units

## Methods

## Process<T>(IObservable<T>)

Sends a request to get a specific setting for an axis.

```
public override IObservable<double> Process<T>(IObservable<T> source)
```

Parameters

source [IObservable<T>](#)

Returns

[IObservable<double>](#)

On each event, it will emit the response of the device.

Type Parameters

T

# Class Home

Namespace: [AllenNeuralDynamics.Zaber](#)

Assembly: AllenNeuralDynamics.Zaber.dll

Represents an operator that instructs a [ZaberDevice](#) to perform a homing routine.

```
public class Home : Sink
```

## Inheritance

[object](#) ← [Combinator](#) ← [Sink](#) ← Home

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Axis

Gets or sets the axis of the manipulator to be controlled.

```
public int? Axis { get; set; }
```

## Property Value

[int](#)?

## Device

Gets or sets the device to be controlled. Defaults to 0.

```
public int Device { get; set; }
```

## Property Value

## PortName

Gets or sets the COM port or alias of the target [ZaberDevice](#)

```
[TypeConverter(typeof(PortNameConverter))]  
public string PortName { get; set; }
```

### Property Value

[string↗](#)

## Methods

### Process<TSource>(IObservable<TSource>)

Homes the target axis when an event is received.

```
public override IObservable<TSource> Process<TSource>(IObservable<TSource> source)
```

### Parameters

**source** [IObservable↗<TSource>](#)

### Returns

[IObservable↗<TSource>](#)

Returns the original input sequence.

### Type Parameters

**TSource**

# Class IsBusy

Namespace: [AllenNeuralDynamics.Zaber](#)

Assembly: AllenNeuralDynamics.Zaber.dll

Represents an operator queries the current state of a [ZaberDevice](#) axis.

```
public class IsBusy : Source<bool>
```

## Inheritance

[object](#) ← [Source](#)<[bool](#)> ← IsBusy

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Axis

Gets or sets the axis of the manipulator to be controlled.

```
public int? Axis { get; set; }
```

## Property Value

[int](#)?

## Device

Gets or sets the device to be controlled. Defaults to 0.

```
public int Device { get; set; }
```

## Property Value

## PortName

Gets or sets the COM port or alias of the target [ZaberDevice](#)

```
[TypeConverter(typeof(PortNameConverter))]  
public string PortName { get; set; }
```

### Property Value

[string ↗](#)

## Methods

### Generate()

Queries the current state of an axis from the manipulator.

```
public override IObservable<bool> Generate()
```

### Returns

[IObservable<bool>](#)

When subscribed, it will emit a sequence with a single value of the most axis state.

### Generate<TSource>(IObservable<TSource>)

Queries the current state of an axis from the manipulator on each event.

```
public IObservable<bool> Generate<TSource>(IObservable<TSource> source)
```

### Parameters

source [IObservable<TSource>](#)

## Returns

[IObservable](#)<[bool](#)>

On each event, it will emit a value with a single value of the most recent axis state.

## Type Parameters

**TSource**

# Class MoveAbsolute

Namespace: [AllenNeuralDynamics.Zaber](#)

Assembly: AllenNeuralDynamics.Zaber.dll

Represents an operator that instructs a [ZaberDevice](#) move to the target absolute position.

```
public class MoveAbsolute : Sink<double>
```

## Inheritance

[object](#) ← [Combinator](#)<[double](#), [double](#)> ← [Sink](#)<[double](#)> ← MoveAbsolute

## Inherited Members

[Combinator](#)<[double](#), [double](#)>.Process([IObservable](#)<[double](#)>), [object.ToString\(\)](#),  
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),  
[object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

# Properties

## Acceleration

Gets or sets optional acceleration of the movement.

```
public double? Acceleration { get; set; }
```

## Property Value

[double](#)?

## AccelerationUnits

Gets or sets the Units of acceleration the manipulator instruction is operating on.

```
[TypeConverter(typeof(AccelerationUnitsConverter))]  
public Units AccelerationUnits { get; set; }
```

Property Value

Units

## Axis

Gets or sets the axis of the manipulator to be controlled.

```
public int Axis { get; set; }
```

Property Value

[int](#)

## Device

Gets or sets the device to be controlled. Defaults to 0.

```
public int Device { get; set; }
```

Property Value

[int](#)

## PortName

Gets or sets the COM port or alias of the target [ZaberDevice](#)

```
[TypeConverter(typeof(PortNameConverter))]  
public string PortName { get; set; }
```

Property Value

[string](#)

## Units

Gets or sets the Units of position the manipulator instruction is operating on.

```
[TypeConverter(typeof(PositionUnitsConverter))]  
public Units Units { get; set; }
```

### Property Value

Units

## Velocity

Gets or sets optional velocity of the movement.

```
public double? Velocity { get; set; }
```

### Property Value

[double](#)?

## VelocityUnits

Gets or sets the Units of velocity the manipulator instruction is operating on.

```
[TypeConverter(typeof(VelocityUnitsConverter))]  
public Units VelocityUnits { get; set; }
```

### Property Value

Units

## Methods

### Process(IObservable<double>)

Moves to the target absolute position when a valid value is received.

```
public override IObservable<double> Process(IObservable<double> source)
```

## Parameters

source [IObservable<double>](#)

## Returns

[IObservable<double>](#)

Returns the original input sequence.

# Class MoveRelative

Namespace: [AllenNeuralDynamics.Zaber](#)

Assembly: AllenNeuralDynamics.Zaber.dll

Represents an operator that instructs a [ZaberDevice](#) move to a relative position.

```
public class MoveRelative : Sink<double>
```

## Inheritance

[object](#) ← [Combinator](#)<[double](#), [double](#)> ← [Sink](#)<[double](#)> ← MoveRelative

## Inherited Members

[Combinator](#)<[double](#), [double](#)>.Process([IObservable](#)<[double](#)>), [object.ToString\(\)](#),  
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),  
[object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

# Properties

## Acceleration

Gets or sets optional acceleration of the movement.

```
public double? Acceleration { get; set; }
```

## Property Value

[double](#)?

## AccelerationUnits

Gets or sets the Units of acceleration the manipulator instruction is operating on.

```
[TypeConverter(typeof(AccelerationUnitsConverter))]  
public Units AccelerationUnits { get; set; }
```

Property Value

Units

## Axis

Gets or sets the axis of the manipulator to be controlled.

```
public int Axis { get; set; }
```

Property Value

[int](#)

## Device

Gets or sets the device to be controlled. Defaults to 0.

```
public int Device { get; set; }
```

Property Value

[int](#)

## PortName

Gets or sets the COM port or alias of the target [ZaberDevice](#)

```
[TypeConverter(typeof(PortNameConverter))]  
public string PortName { get; set; }
```

Property Value

[string](#)

## Units

Gets or sets the Units of position the manipulator instruction is operating on.

```
[TypeConverter(typeof(PositionUnitsConverter))]  
public Units Units { get; set; }
```

### Property Value

Units

## Velocity

Gets or sets optional velocity of the movement.

```
public double? Velocity { get; set; }
```

### Property Value

[double](#)?

## VelocityUnits

Gets or sets the Units of velocity the manipulator instruction is operating on.

```
[TypeConverter(typeof(VelocityUnitsConverter))]  
public Units VelocityUnits { get; set; }
```

### Property Value

Units

## Methods

### Process(IObservable<double>)

Moves to the target relative position when a valid value is received.

```
public override IObservable<double> Process(IObservable<double> source)
```

## Parameters

source [IObservable<double>](#)

## Returns

[IObservable<double>](#)

Returns the original input sequence.

# Class MoveVelocity

Namespace: [AllenNeuralDynamics.Zaber](#)

Assembly: AllenNeuralDynamics.Zaber.dll

Represents an operator that instructs a [ZaberDevice](#) move with a given velocity value.

```
public class MoveVelocity : Sink<double>
```

## Inheritance

[object](#) ← [Combinator](#)<[double](#), [double](#)> ← [Sink](#)<[double](#)> ← MoveVelocity

## Inherited Members

[Combinator](#)<[double](#), [double](#)>.Process([IObservable](#)<[double](#)>), [object.ToString\(\)](#),  
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),  
[object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

## Properties

### Acceleration

Gets or sets optional acceleration of the movement.

```
public double? Acceleration { get; set; }
```

### Property Value

[double](#)?

### AccelerationUnits

Gets or sets the Units of acceleration the manipulator instruction is operating on.

```
[TypeConverter(typeof(AccelerationUnitsConverter))]  
public Units AccelerationUnits { get; set; }
```

Property Value

Units

## Axis

Gets or sets the axis of the manipulator to be controlled.

```
public int Axis { get; set; }
```

Property Value

[int](#)

## Device

Gets or sets the device to be controlled. Defaults to 0.

```
public int Device { get; set; }
```

Property Value

[int](#)

## PortName

Gets or sets the COM port or alias of the target [ZaberDevice](#)

```
[TypeConverter(typeof(PortNameConverter))]  
public string PortName { get; set; }
```

Property Value

[string](#)

# VelocityUnits

Gets or sets the Units of velocity the manipulator instruction is operating on.

```
[TypeConverter(typeof(VelocityUnitsConverter))]  
public Units VelocityUnits { get; set; }
```

Property Value

Units

## Methods

### Process(IObservable<double>)

Instructs the manipulator axis to move at a constant specified velocity.

```
public override IObservable<double> Process(IObservable<double> source)
```

Parameters

source [IObservable<double>](#)

Returns

[IObservable<double>](#)

Returns the original input sequence.

# Class Park

Namespace: [AllenNeuralDynamics.Zaber](#)

Assembly: AllenNeuralDynamics.Zaber.dll

Represents an operator that instructs a [ZaberDevice](#) to park, or unpark, a selected axis or device.

```
public class Park : Sink<bool>
```

## Inheritance

[object](#) ← [Combinator](#)<[bool](#), [bool](#)> ← [Sink](#)<[bool](#)> ← Park

## Inherited Members

[Combinator](#)<[bool](#), [bool](#)>.Process([IObservable](#)<[bool](#)>), [object.ToString\(\)](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),  
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

# Properties

## Axis

Gets or sets the axis of the manipulator to be controlled.

```
public int? Axis { get; set; }
```

## Property Value

[int](#)?

## Device

Gets or sets the device to be controlled. Defaults to 0.

```
public int Device { get; set; }
```

## Property Value

## PortName

Gets or sets the COM port or alias of the target [ZaberDevice](#)

```
[TypeConverter(typeof(PortNameConverter))]  
public string PortName { get; set; }
```

### Property Value

[string↗](#)

## Methods

### Process(IObservable<bool>)

Parks (True), or Unparks(False) a specified axis or device when a value is received.

```
public override IObservable<bool> Process(IObservable<bool> source)
```

### Parameters

**source** [IObservable↗ <bool↗ >](#)

### Returns

[IObservable↗ <bool↗ >](#)

Returns the original bool input sequence.

# Class PositionUnitsConverter

Namespace: [AllenNeuralDynamics.Zaber](#)

Assembly: AllenNeuralDynamics.Zaber.dll

```
public class PositionUnitsConverter : EnumConverter
```

## Inheritance

[object](#) ← [TypeConverter](#) ← [EnumConverter](#) ← PositionUnitsConverter

## Inherited Members

[EnumConverter.CanConvertFrom\(ITypeDescriptorContext, Type\)](#) ,  
[EnumConverter.CanConvertTo\(ITypeDescriptorContext, Type\)](#) ,  
[EnumConverter.ConvertFrom\(ITypeDescriptorContext, CultureInfo, object\)](#) ,  
[EnumConverter.ConvertTo\(ITypeDescriptorContext, CultureInfo, object, Type\)](#) ,  
[EnumConverter.GetStandardValuesExclusive\(ITypeDescriptorContext\)](#) ,  
[EnumConverter.GetStandardValuesSupported\(ITypeDescriptorContext\)](#) ,  
[EnumConverter.IsValid\(ITypeDescriptorContext, object\)](#) , [EnumConverter.EnumType](#) ,  
[EnumConverter.Values](#) , [EnumConverter.Comparer](#) , [TypeConverter.CanConvertFrom\(Type\)](#) ,  
[TypeConverter.CanConvertTo\(Type\)](#) , [TypeConverter.ConvertFrom\(object\)](#) ,  
[TypeConverter.ConvertFromInvariantString\(string\)](#) ,  
[TypeConverter.ConvertFromInvariantString\(ITypeDescriptorContext, string\)](#) ,  
[TypeConverter.ConvertFromString\(string\)](#) ,  
[TypeConverter.ConvertFromString\(ITypeDescriptorContext, string\)](#) ,  
[TypeConverter.ConvertFromString\(ITypeDescriptorContext, CultureInfo, string\)](#) ,  
[TypeConverter.ConvertTo\(object, Type\)](#) , [TypeConverter.ConvertToInvariantString\(object\)](#) ,  
[TypeConverter.ConvertToInvariantString\(ITypeDescriptorContext, object\)](#) ,  
[TypeConverter.ConvertToString\(object\)](#) ,  
[TypeConverter.ConvertToString\(ITypeDescriptorContext, object\)](#) ,  
[TypeConverter.ConvertToString\(ITypeDescriptorContext, CultureInfo, object\)](#) ,  
[TypeConverter.CreateInstance\(IDictionary\)](#) ,  
[TypeConverter.CreateInstance\(ITypeDescriptorContext, IDictionary\)](#) ,  
[TypeConverter.GetConvertFromException\(object\)](#) ,  
[TypeConverter.GetConvertToException\(object, Type\)](#) , [TypeConverter.GetCreateInstanceSupported\(\)](#) ,  
[TypeConverter.GetCreateInstanceSupported\(ITypeDescriptorContext\)](#) ,  
[TypeConverter.GetProperties\(object\)](#) , [TypeConverter.GetProperties\(ITypeDescriptorContext, object\)](#) ,  
[TypeConverter.GetProperties\(ITypeDescriptorContext, object, Attribute\[\]\)](#) ,  
[TypeConverter.GetPropertiesSupported\(\)](#) ,  
[TypeConverter.GetPropertiesSupported\(ITypeDescriptorContext\)](#) ,

[TypeConverter.GetStandardValues\(\)](#) , [TypeConverter.GetStandardValuesExclusive\(\)](#) ,  
[TypeConverter.GetStandardValuesSupported\(\)](#) , [TypeConverter.IsValid\(object\)](#) ,  
[TypeConverter.SortProperties\(PropertyDescriptorCollection, string\[\]\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### PositionUnitsConverter()

```
public PositionUnitsConverter()
```

## Methods

### GetStandardValues(ITypeDescriptorContext)

Gets a collection of standard values for the data type this validator is designed for.

```
public override TypeConverter.StandardValuesCollection  
GetStandardValues(ITypeDescriptorContext context)
```

#### Parameters

context [ITypeDescriptorContext](#)

An [ITypeDescriptorContext](#) that provides a format context.

#### Returns

[TypeConverter](#).[StandardValuesCollection](#)

A [TypeConverter.StandardValuesCollection](#) that holds a standard set of valid values, or [null](#) if the data type does not support a standard set of values.

# Class SetSetting

Namespace: [AllenNeuralDynamics.Zaber](#)

Assembly: AllenNeuralDynamics.Zaber.dll

Represents an operator that sets a setting for a specified axis.

```
public class SetSetting : Sink<double>
```

## Inheritance

[object](#) ← [Combinator](#)<[double](#), [double](#)> ← [Sink](#)<[double](#)> ← SetSetting

## Inherited Members

[Combinator<double, double>.Process\(IObservable<double>\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

# Properties

## Axis

Gets or sets the axis of the manipulator to be controlled.

```
public int Axis { get; set; }
```

## Property Value

[int](#)

## Device

Gets or sets the device to be controlled. Defaults to 0.

```
public int? Device { get; set; }
```

## Property Value

[int](#)?

## PortName

Gets or sets the COM port or alias of the target [ZaberDevice](#)

```
[TypeConverter(typeof(PortNameConverter))]  
public string PortName { get; set; }
```

### Property Value

[string](#)

## Setting

Gets or sets the axis setting to be set.

```
public string Setting { get; set; }
```

### Property Value

[string](#)

## Units

Gets or sets the Units of the transaction.

```
public Units Units { get; set; }
```

### Property Value

Units

## Methods

## Process(I<sup>O</sup>bservable<double>)

Moves to the target absolute position when a valid value is received.

```
public override IObservable<double> Process(IObservable<double> source)
```

### Parameters

source [I<sup>O</sup>bservable<double>](#)

### Returns

[I<sup>O</sup>bservable<double>](#)

Returns the original input sequence.

# Class Stop

Namespace: [AllenNeuralDynamics.Zaber](#)

Assembly: AllenNeuralDynamics.Zaber.dll

Represents an operator that instructs a [ZaberDevice](#) to stop any movement on a selected axis.

```
public class Stop : Sink
```

## Inheritance

[object](#) ← [Combinator](#) ← [Sink](#) ← Stop

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Axis

Gets or sets the axis of the manipulator to be controlled.

```
public int? Axis { get; set; }
```

## Property Value

[int](#)?

## Device

Gets or sets the device to be controlled. Defaults to 0.

```
public int Device { get; set; }
```

## Property Value

## PortName

Gets or sets the COM port or alias of the target [ZaberDevice](#)

```
[TypeConverter(typeof(PortNameConverter))]  
public string PortName { get; set; }
```

### Property Value

[string↗](#)

## Methods

### Process<TSource>(IObservable<TSource>)

Halts the movement of a specified axis when a value is received.

```
public override IObservable<TSource> Process<TSource>(IObservable<TSource> source)
```

### Parameters

source [IObservable↗<TSource>](#)

### Returns

[IObservable↗<TSource>](#)

Returns the original input sequence.

### Type Parameters

[TSource](#)

# Class VelocityUnitsConverter

Namespace: [AllenNeuralDynamics.Zaber](#)

Assembly: AllenNeuralDynamics.Zaber.dll

```
public class VelocityUnitsConverter : EnumConverter
```

## Inheritance

[object](#) ← [TypeConverter](#) ← [EnumConverter](#) ← VelocityUnitsConverter

## Inherited Members

[EnumConverter.CanConvertFrom\(ITypeDescriptorContext, Type\)](#) ,  
[EnumConverter.CanConvertTo\(ITypeDescriptorContext, Type\)](#) ,  
[EnumConverter.ConvertFrom\(ITypeDescriptorContext, CultureInfo, object\)](#) ,  
[EnumConverter.ConvertTo\(ITypeDescriptorContext, CultureInfo, object, Type\)](#) ,  
[EnumConverter.GetStandardValuesExclusive\(ITypeDescriptorContext\)](#) ,  
[EnumConverter.GetStandardValuesSupported\(ITypeDescriptorContext\)](#) ,  
[EnumConverter.IsValid\(ITypeDescriptorContext, object\)](#) , [EnumConverter.EnumType](#) ,  
[EnumConverter.Values](#) , [EnumConverter.Comparer](#) , [TypeConverter.CanConvertFrom\(Type\)](#) ,  
[TypeConverter.CanConvertTo\(Type\)](#) , [TypeConverter.ConvertFrom\(object\)](#) ,  
[TypeConverter.ConvertFromInvariantString\(string\)](#) ,  
[TypeConverter.ConvertFromInvariantString\(ITypeDescriptorContext, string\)](#) ,  
[TypeConverter.ConvertFromString\(string\)](#) ,  
[TypeConverter.ConvertFromString\(ITypeDescriptorContext, string\)](#) ,  
[TypeConverter.ConvertFromString\(ITypeDescriptorContext, CultureInfo, string\)](#) ,  
[TypeConverter.ConvertTo\(object, Type\)](#) , [TypeConverter.ConvertToInvariantString\(object\)](#) ,  
[TypeConverter.ConvertToInvariantString\(ITypeDescriptorContext, object\)](#) ,  
[TypeConverter.ConvertToString\(object\)](#) ,  
[TypeConverter.ConvertToString\(ITypeDescriptorContext, object\)](#) ,  
[TypeConverter.ConvertToString\(ITypeDescriptorContext, CultureInfo, object\)](#) ,  
[TypeConverter.CreateInstance\(IDictionary\)](#) ,  
[TypeConverter.CreateInstance\(ITypeDescriptorContext, IDictionary\)](#) ,  
[TypeConverter.GetConvertFromException\(object\)](#) ,  
[TypeConverter.GetConvertToException\(object, Type\)](#) , [TypeConverter.GetCreateInstanceSupported\(\)](#) ,  
[TypeConverter.GetCreateInstanceSupported\(ITypeDescriptorContext\)](#) ,  
[TypeConverter.GetProperties\(object\)](#) , [TypeConverter.GetProperties\(ITypeDescriptorContext, object\)](#) ,  
[TypeConverter.GetProperties\(ITypeDescriptorContext, object, Attribute\[\]\)](#) ,  
[TypeConverter.GetPropertiesSupported\(\)](#) ,  
[TypeConverter.GetPropertiesSupported\(ITypeDescriptorContext\)](#) ,

[TypeConverter.GetStandardValues\(\)](#) , [TypeConverter.GetStandardValuesExclusive\(\)](#) ,  
[TypeConverter.GetStandardValuesSupported\(\)](#) , [TypeConverter.IsValid\(object\)](#) ,  
[TypeConverter.SortProperties\(PropertyDescriptorCollection, string\[\]\)](#) , [object.ToString\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### VelocityUnitsConverter()

```
public VelocityUnitsConverter()
```

## Methods

### GetStandardValues(ITypeDescriptorContext)

Gets a collection of standard values for the data type this validator is designed for.

```
public override TypeConverter.StandardValuesCollection  
GetStandardValues(ITypeDescriptorContext context)
```

#### Parameters

**context** [ITypeDescriptorContext](#)

An [ITypeDescriptorContext](#) that provides a format context.

#### Returns

[TypeConverter](#).[StandardValuesCollection](#)

A [TypeConverter.StandardValuesCollection](#) that holds a standard set of valid values, or [null](#) if the data type does not support a standard set of values.

# Class WaitUntilIdle

Namespace: [AllenNeuralDynamics.Zaber](#)

Assembly: AllenNeuralDynamics.Zaber.dll

Represents an operator that queries and waits to return until a [ZaberDevice](#) goes idle.

```
public class WaitUntilIdle : Combinator<Unit>
```

## Inheritance

[object](#) ← [Combinator](#)<[Unit](#)> ← WaitUntilIdle

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

# Properties

## Axis

Gets or sets the axis of the manipulator to be controlled.

```
public int? Axis { get; set; }
```

## Property Value

[int](#)?

## Device

Gets or sets the device to be controlled. Defaults to 0.

```
public int Device { get; set; }
```

## Property Value

## PortName

Gets or sets the COM port or alias of the target [ZaberDevice](#)

```
[TypeConverter(typeof(PortNameConverter))]  
public string PortName { get; set; }
```

## Property Value

[string↗](#)

## Methods

### Generate()

Queries the idle state of the manipulator

```
public IObservable<Unit> Generate()
```

## Returns

[IObservable↗ <Unit↗ >](#)

When subscribed, it will emit a sequence with a single value when the manipulator is idle.

### Process<TSource>(IObservable<TSource>)

Queries the idle state of the manipulator

```
public override IObservable<Unit> Process<TSource>(IObservable<TSource> source)
```

## Parameters

source [IObservable↗ <TSource>](#)

## Returns

[IObservable](#) <Unit>

On each event, it will emit a sequence with a single value when the manipulator is idle.

## Type Parameters

TSource

# Class ZaberDevice

Namespace: [AllenNeuralDynamics.Zaber](#)

Assembly: AllenNeuralDynamics.Zaber.dll

Represents an Zaber manipulator communicating with the host computer using the Zaber.Motion SDK.

```
public sealed class ZaberDevice : IDisposable
```

## Inheritance

[object](#) ← ZaberDevice

## Implements

[IDisposable](#)

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#)

## Constructors

### ZaberDevice(string)

Initializes a new instance of the [ZaberDevice](#) class using the specified port name.

```
public ZaberDevice(string portName)
```

## Parameters

`portName` [string](#)

The port to use (for example, COM1).

## Properties

### IsOpen

```
public bool IsOpen { get; }
```

Property Value

[bool](#)

## Methods

### Close()

Closes the port connection, sets the [IsOpen](#) property to [false](#) and disposes of the internal Zaber.Motion.Ascii.Connection object.

```
public void Close()
```

### GenericCommand(int?, int?, string)

```
public Task<Response> GenericCommand(int? deviceIndex, int? axis, string command)
```

Parameters

deviceIndex [int](#)?

axis [int](#)?

command [string](#)

Returns

[Task](#) <Response>

### GenericCommandMultiResponse(int?, int?, string)

```
public Task<Response[]> GenericCommandMultiResponse(int? deviceIndex, int? axis, string command)
```

Parameters

deviceIndex [int](#)?

axis [int](#)?

command [string](#)

Returns

[Task](#)<Response[]>

## GenericCommandNoResponse(int?, int?, string)

```
public void GenericCommandNoResponse(int? deviceIndex, int? axis, string command)
```

Parameters

deviceIndex [int](#)?

axis [int](#)?

command [string](#)

## GetPosition(int?, int, Units)

```
public Task<double> GetPosition(int? deviceIndex, int axis, Units unit)
```

Parameters

deviceIndex [int](#)?

axis [int](#)

unit Units

Returns

[Task](#)<[double](#)>

## GetSetting(int?, int, string, Units)

```
public Task<double> GetSetting(int? deviceIndex, int axis, string setting, Units unit)
```

### Parameters

deviceIndex [int](#)?

axis [int](#)

setting [string](#)

unit Units

### Returns

[Task](#)<[double](#)>

## Home(int?, int?)

```
public void Home(int? deviceIndex, int? axis)
```

### Parameters

deviceIndex [int](#)?

axis [int](#)?

## IsBusy(int?, int?)

```
public Task<bool> IsBusy(int? deviceIndex, int? axis)
```

### Parameters

deviceIndex [int](#)?

`axis` [int?](#)

Returns

[Task](#) <[bool](#)>

**MoveAbsolute(int?, int, double, double, double, Units, Units, Units)**

```
public void MoveAbsolute(int? deviceIndex, int axis, double position, double velocity,
double acceleration, Units unit, Units velocityUnit, Units accelerationUnit)
```

Parameters

`deviceIndex` [int?](#)

`axis` [int](#)

`position` [double](#)

`velocity` [double](#)

`acceleration` [double](#)

`unit` [Units](#)

`velocityUnit` [Units](#)

`accelerationUnit` [Units](#)

**MoveRelative(int?, int, double, double, double, Units, Units, Units)**

```
public void MoveRelative(int? deviceIndex, int axis, double position, double velocity,
double acceleration, Units unit, Units velocityUnit, Units accelerationUnit)
```

Parameters

`deviceIndex` [int?](#)

`axis` [int](#)

`position` [double](#)

`velocity` [double](#)

`acceleration` [double](#)

`unit` Units

`velocityUnit` Units

`accelerationUnit` Units

## MoveVelocity(int?, int, double, double, Units, Units)

```
public void MoveVelocity(int? deviceIndex, int axis, double velocity, double acceleration,  
    Units velocityUnit, Units accelerationUnit)
```

### Parameters

`deviceIndex` [int](#)?

`axis` [int](#)

`velocity` [double](#)

`acceleration` [double](#)

`velocityUnit` Units

`accelerationUnit` Units

## Open()

Opens a new connection to a [ZaberDevice](#).

```
public void Open()
```

## Park(int?, int?)

```
public void Park(int? deviceIndex, int? axis)
```

### Parameters

deviceIndex [int↗?](#)

axis [int↗?](#)

## SetSetting(int?, int, string, double, Units)

```
public void SetSetting(int? deviceIndex, int axis, string setting, double value, Units units  
= Units.Native)
```

### Parameters

deviceIndex [int↗?](#)

axis [int↗](#)

setting [string↗](#)

value [double↗](#)

units Units

## Stop(int?, int?)

```
public void Stop(int? deviceIndex, int? axis)
```

### Parameters

deviceIndex [int↗?](#)

axis [int↗?](#)

## Unpark(int?, int?)

```
public void Unpark(int? deviceIndex, int? axis)
```

### Parameters

deviceIndex [int↗?](#)

axis [int↗?](#)

## WaitUntilIdle(int?, int?)

```
public Task<Unit> WaitUntilIdle(int? deviceIndex, int? axis)
```

### Parameters

deviceIndex [int↗?](#)

axis [int↗?](#)

### Returns

[Task↗<Unit↗>](#)

# Class ZaberDeviceConfiguration

Namespace: [AllenNeuralDynamics.Zaber](#)

Assembly: AllenNeuralDynamics.Zaber.dll

Represents configuration settings used to initialize a connection with a [ZaberDevice](#)

```
public class ZaberDeviceConfiguration
```

## Inheritance

[object](#) ← ZaberDeviceConfiguration

## Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#)

## Constructors

### ZaberDeviceConfiguration()

Initializes a new instance of the [ZaberDeviceConfiguration](#) class.

```
public ZaberDeviceConfiguration()
```

## Properties

### PortName

Gets or sets the name of the serial port.

```
[TypeConverter(typeof(SerialPortNameConverter))]  
public string PortName { get; set; }
```

### Property Value

[string ↗](#)