Namespace KIRSharp

Classes

Broadcast

Broadcast singleton class

Broadcast.Info

Broadcast camera information class

FrameBuffer

Classes to store camera frame buffers.

UdpCamera

Udp Camera partial class

Util

Utility static methods

Interfaces

ICamera

Camera method interface

IFrame

Camera frame callback argument

Enums

Broadcast.Info.SensorType

Broadcast camera sensor types enum

ICamera.CameraType

Camera types enum

IFrame.FrameType

Frame types enum

Class Broadcast

Namespace: KIRSharp

Assembly: KIRSharp.dll

Broadcast singleton class

```
public class Broadcast : IDisposable
```

Inheritance

object ← Broadcast

Implements

IDisposable

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

Instance

Broadcast singleton class instance

```
public static Broadcast Instance { get; }
```

Property Value

Broadcast

Methods

Dispose()

Dispose

```
public void Dispose()
```

FindCamera()

Find broadcast camera

```
public Task<Broadcast.Info?> FindCamera()
```

Returns

Task<Broadcast.Info>

Class Broadcast.Info

Namespace: KIRSharp

Assembly: KIRSharp.dll

Broadcast camera information class

```
public class Broadcast.Info
```

Inheritance

object ← Broadcast.Info

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

Info()

Constructor

```
public Info()
```

Info(Info)

Constructor

```
public Info(Broadcast.Info info)
```

Parameters

`info` Broadcast.Info

Info(byte[])

Constructor

```
public Info(byte[] receiveBuffer)
```

Parameters

`receiveBuffer` byte[]

Info(string)

Constructor

```
public Info(string receiveMessage)
```

Parameters

`receiveMessage` string

Properties

Connections

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

Property Value

int

CustomName

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

Property Value

string

DBVT

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

Property Value

int

Description

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

Property Value

string

Flip

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

Property Value

int

Gateway

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

Property Value

string

Ip

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

Property Value

string

Name

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

Property Value

string

NetMask

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

Property Value

string

Opm

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

Property Value

int

Port

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

Property Value

string

RtspCMOS

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

Property Value

string

RtspIR

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

Property Value

string

SEC

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

Property Value

string

Sens

```
public Broadcast.Info.SensorType Sens { get; set; }
```

Property Value

Broadcast.Info.SensorType

SiteName

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

Property Value

string

UVF

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

Property Value

int

Version

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

Property Value

string

WatchDog

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

Property Value

int

Enum Broadcast.Info.SensorType

Namespace: KIRSharp

Assembly: KIRSharp.dll

Broadcast camera sensor types enum

```
public enum Broadcast.Info.SensorType
```

Fields

`IR_SENSOR_160 = 2`

`IR_SENSOR_320 = 3`

`IR_SENSOR_384 = 4`

`IR_SENSOR_80 = 0`

`IR_SENSOR_80_SHUTTER = 1`

`Unknown = 5`

Class FrameBuffer

Namespace: KIRSharp

Assembly: KIRSharp.dll

Classes to store camera frame buffers.

```
public class FrameBuffer
```

Inheritance

object ← FrameBuffer

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

FrameBuffer(int, int)

Constructor

```
public FrameBuffer(int fps, int period = 60)
```

Parameters

`fps` int

`period` int

FrameBuffer(int, int, int, int)

Constructor

```
public FrameBuffer(int width, int height, int fps, int period = 60)
```

Parameters

`width` int

`height` int

`fps` int

`period` int

Properties

Fps

Camera fps

```
public int Fps { get; }
```

Property Value

int

FrameHeight

Frame Height

```
public int FrameHeight { get; }
```

Property Value

int

FrameWidth

Frame width

```
public int FrameWidth { get; }
```

Property Value

int

Period

Buffer storage duration

```
public int Period { get; }
```

Property Value

int

Queue

Frame data buffer queue

```
public ConcurrentQueue<byte[]> Queue { get; }
```

Property Value

ConcurrentQueue<byte[]>

Size

buffer size

```
public int Size { get; }
```

Property Value

int

Methods

Enqueue(byte[])

Enqueue frame to buffer

```
public Task Enqueue(byte[] buffer)
```

Parameters

`buffer` byte[]

Returns

Task

SetWidthHeight(int, int)

Set frame width and height

```
public void SetWidthHeight(int width, int height)
```

Parameters

`width` int

`height` int

Interface ICamera

Namespace: KIRSharp

Assembly: KIRSharp.dll

Camera method interface

```
public interface ICamera : IDisposable
```

Inherited Members

IDisposable.Dispose()

Properties

FrameBufferCmos

Classes to store information about cmos camera frames and buffer queues

```
FrameBuffer FrameBufferCmos { get; }
```

Property Value

FrameBuffer

FrameBufferThermal

Classes to store information about thermal camera frames and buffer queues

```
FrameBuffer FrameBufferThermal { get; }
```

Property Value

FrameBuffer

Info

Classes with Broadcast Information

```
Broadcast.Info Info { get; }
```

Property Value

Broadcast.Info

IsRun

```
bool IsRun { get; set; }
```

Property Value

bool

Type

Camera type

```
ICamera.CameraType Type { get; }
```

Property Value

ICamera.CameraType

UvfCount

Value of the UV sensor

```
short UvfCount { get; }
```

Property Value

short

Methods

OnCmosFrameEnqueued(FrameEventArgs)

Cmos frame callback

```
void OnCmosFrameEnqueued(FrameEventArgs e)
```

Parameters

e FrameEventArgs

OnThermalFrameEnqueued(FrameEventArgs)

Thermal frame callback

```
void OnThermalFrameEnqueued(FrameEventArgs e)
```

Parameters

e FrameEventArgs

OnUvfCountEnqueued(FrameEventArgs)

Uv sensor callback

```
void OnUvfCountEnqueued(FrameEventArgs e)
```

Parameters

e FrameEventArgs

StartStreaming(int, int)

Start camera streaming

```
void StartStreaming(int fpsThermal, int fpsCmos)
```

Parameters

`fpsThermal` int

`fpsCmos` int

StopStreamingAsync()

Stop camera streaming

```
Task StopStreamingAsync()
```

Returns

Task

Events

CmosFrameEnqueued

Event callback returning the FrameEventArgs of the cmos camera

```
event EventHandler<FrameEventArgs> CmosFrameEnqueued
```

Event Type

EventHandler<FrameEventArgs>

ThermalFrameEnqueued

Event callback returning the FrameEventArgs of the thermal camera

```
event EventHandler<FrameEventArgs> ThermalFrameEnqueued
```

Event Type

EventHandler<FrameEventArgs>

UvfCountEnqueued

Event callback returning the FrameEventArgs of the uv sensor

```
event EventHandler<FrameEventArgs> UvfCountEnqueued
```

Event Type

EventHandler<FrameEventArgs>

Enum ICamera.CameraType

Namespace: KIRSharp

Assembly: KIRSharp.dll

Camera types enum

```
public enum ICamera.CameraType
```

Fields

`KIR160_Kelvin = 2`

`KIR160_Raw = 1`

`KIR256_Kelvin = 3`

`KIR384_Kelvin = 4`

`KIR80_Kelvin = 0`

`TestDummy = 5`

Interface IFrame

Namespace: KIRSharp

Assembly: KIRSharp.dll

Camera frame callback argument

```
public interface IFrame
```

Properties

Bytes

Frame bytes buffer

```
byte[] Bytes { get; }
```

Property Value

byte[]

Height

Frame Height

```
int Height { get; }
```

Property Value

int

Type

Camera frame type

```
IFrame.FrameType Type { get; }
```

Property Value

IFrame.FrameType

Width

Frame width

```
int Width { get; }
```

Property Value

int

Methods

QueryAsync(byte[])

```
Task<bool> QueryAsync(byte[] receivedPacket)
```

Parameters

`receivedPacket` byte[]

Returns

Task<bool>

Enum IFrame.FrameType

Namespace: KIRSharp

Assembly: KIRSharp.dll

Frame types enum

```
public enum IFrame.FrameType
```

Fields

`Cmos = 1`

`Thermal = 0`

Class UdpCamera

Namespace: KIRSharp

Assembly: KIRSharp.dll

Udp Camera partial class

```
public class UdpCamera : ObservableObject, INotifyPropertyChanged, INotifyPropertyChanging,
ICamera, IDisposable
```

Inheritance

object ← ObservableObject ← UdpCamera

Implements

INotifyPropertyChanged, INotifyPropertyChanging, ICamera, IDisposable

Inherited Members

ObservableObject.OnPropertyChanged(PropertyChangedEventArgs) ,
ObservableObject.OnPropertyChanging(PropertyChangingEventArgs) ,
ObservableObject.OnPropertyChanged(string) , ObservableObject.OnPropertyChanging(string) ,
ObservableObject.SetProperty<T>(ref T, T, string) ,
ObservableObject.SetProperty<T>(ref T, T, IEqualityComparer<T>, string) ,
ObservableObject.SetProperty<T>(T, T, Action<T>, string) ,
ObservableObject.SetProperty<T>(T, T, IEqualityComparer<T>, Action<T>, string) ,
ObservableObject.SetProperty<TModel, T>(T, T, TModel, Action<TModel, T>, string) ,
ObservableObject.SetProperty<TModel, T>(T, T, IEqualityComparer<T>, TModel, Action<TModel, T>, string) ,
ObservableObject.SetPropertyAndNotifyOnCompletion(ref ObservableObject.TaskNotifier, Task, string) ,
ObservableObject.SetPropertyAndNotifyOnCompletion(ref ObservableObject.TaskNotifier, Task, Action<Task>, string) ,
ObservableObject.SetPropertyAndNotifyOnCompletion<T>(ref ObservableObject.TaskNotifier<T>, Task<T>, string) ,
ObservableObject.SetPropertyAndNotifyOnCompletion<T>(ref ObservableObject.TaskNotifier<T>, Task<T>, Action<Task<T>>, string) ,
ObservableObject.PropertyChanged , ObservableObject.PropertyChanging ,
object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() ,
object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

UdpCamera(Info, int, int)

Constructor

```
public UdpCamera(Broadcast.Info info, int fpsThermal, int fpsCmos)
```

Parameters

`info` Broadcast.Info

`fpsThermal` int

`fpsCmos` int

Properties

CmosFps

Cmos camera fps

```
public int CmosFps { get; }
```

Property Value

int

CmosPort

```
public int CmosPort { get; }
```

Property Value

int

CommandPort

```
public int CommandPort { get; }
```

Property Value

int

FrameBufferCmos

Cmos camera frame buffer.

```
public FrameBuffer FrameBufferCmos { get; }
```

Property Value

FrameBuffer

FrameBufferThermal

Thermal camera frame buffer.

```
public FrameBuffer FrameBufferThermal { get; }
```

Property Value

FrameBuffer

Info

Broadcast information

```
public Broadcast.Info Info { get; }
```

Property Value

Broadcast.Info

IrFps

Thermal camera fps

```
public int IrFps { get; }
```

Property Value

int

IsRun

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

Property Value

bool

MainPort

```
public int MainPort { get; }
```

Property Value

int

Type

Camera Type

```
public ICamera.CameraType Type { get; set; }
```

Property Value

ICamera.CameraType

UvfCount

Uv sensor value

```
public short UvfCount { get; set; }
```

Property Value

short

Methods

Dispose()

Dispose instance.

```
public void Dispose()
```

GetCmosFpsAsync()

```
public Task<int> GetCmosFpsAsync()
```

Returns

Task<int>

GetOffset()

Gets the thermal camera offset.

```
public Task<double?> GetOffset()
```

Returns

Task<double?>

A Task<TResult> representing the asynchronous operation. The task result contains the offset value as a double if successful; otherwise, null.

OnCmosFrameEnqueued(FrameEventArgs)

Overriding virtual cmos frame callback event methods

```
protected virtual void OnCmosFrameEnqueued(FrameEventArgs e)
```

Parameters

e FrameEventArgs

OnThermalFrameEnqueued(FrameEventArgs)

Overriding virtual thermal frame callback event methods

```
protected virtual void OnThermalFrameEnqueued(FrameEventArgs e)
```

Parameters

e FrameEventArgs

OnUvfCountEnqueued(FrameEventArgs)

Overriding virtual uv sensor callback event methods

```
protected virtual void OnUvfCountEnqueued(FrameEventArgs e)
```

Parameters

`e` FrameEventArgs

RunShutterManually()

Activate the thermal camera shutter once.

```
public Task<byte[]> RunShutterManually()
```

Returns

Task<byte[]>

SetOffset(double)

Sets the thermal camera offset.

```
public Task<double?> SetOffset(double offset)
```

Parameters

`offset` double

Returns

Task<double?>

A Task<TResult> representing the asynchronous operation. The task result contains the offset value as a double if successful; otherwise, null.

StartStreaming(int, int)

Start streaming method.

```
public void StartStreaming(int fpsThermal, int fpsCmos)
```

Parameters

`fpsThermal` int

`fpsCmos` int

StopStreamingAsync()

Stop streaming method.

```
public Task StopStreamingAsync()
```

Returns

Task

Events

CmosFrameEnqueued

Cmos frame callback event.

```
public event EventHandler<FrameEventArgs> CmosFrameEnqueued
```

Event Type

EventHandler<FrameEventArgs>

ThermalFrameEnqueued

Thermal frame callback event.

```
public event EventHandler<FrameEventArgs> ThermalFrameEnqueued
```

Event Type

EventHandler<FrameEventArgs>

UvfCountEnqueued

Uv sensor callback event.

```
public event EventHandler<FrameEventArgs> UvfCountEnqueued
```

Event Type

EventHandler<FrameEventArgs>

Class Util

Namespace: KIRSharp

Assembly: KIRSharp.dll

Utility static methods

```
public class Util
```

Inheritance

object ← Util

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Methods

BytesToShorts(byte[])

Converts bytes to shorts (MSB -> LSB)

```
public static short[] BytesToShorts(byte[] bytes)
```

Parameters

`bytes` byte[]

Returns

short[]

Namespace KIRSharp.Camera

Classes

CameraName

Define camera names

FrameEventArgs

Camera frame event argument

Class CameraName

Namespace: KIRSharp.Camera

Assembly: KIRSharp.dll

Define camera names

```
public class CameraName
```

Inheritance

object ← CameraName

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Fields

Dummy

```
public const string Dummy = "Dummy"
```

Field Value

string

Kir160Dual

```
public const string Kir160Dual = "KAVAS DualCAM"
```

Field Value

string

Kir384Dual

```
public const string Kir384Dual = "LKSamyang SVS-384TW2I"
```

Field Value

string

Kir80Single

```
public const string Kir80Single = "KAVAS IRCAM"
```

Field Value

string

Class FrameEventArgs

Namespace: KIRSharp.Camera

Assembly: KIRSharp.dll

Camera frame event argument

```
public class FrameEventArgs : EventArgs
```

Inheritance

object ← EventArgs ← FrameEventArgs

Inherited Members

EventArgs.Empty , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

FrameEventArgs(IFrame)

Constructor

```
public FrameEventArgs(IFrame frame)
```

Parameters

`frame` IFrame

IFrame

Properties

Frame

Camera frame interface

```
public IFrame Frame { get; }
```

Property Value

IFrame