

MYNT EYE S SDK

2.3.3

制作者 Doxygen 1.8.15

1 MYNT EYE S SDK	1
2 设备数据说明	3
2.1 硬件信息说明	3
2.2 图像参数说明	4
2.3 IMU 参数说明	5
2.4 图像数据说明	5
2.5 IMU 数据说明	6
3 设备控制说明	9
3.1 控制 API 说明	9
3.2 拓展通道说明	11
4 弃用列表	13
5 模块索引	15
5.1 模块	15
6 继承关系索引	17
6.1 类继承关系	17
7 类索引	19
7.1 类列表	19
8 模块说明	21
8.1 Utilities	21
8.1.1 详细描述	21
8.1.2 函数说明	21
8.1.2.1 get_real_exposure_time()	21
8.1.2.2 select()	22
8.1.2.3 select_request()	22
8.2 Enumerations	23
8.2.1 详细描述	24
8.2.2 枚举类型说明	24
8.2.2.1 AddOns	24
8.2.2.2 Capabilities	25
8.2.2.3 Format	25

8.2.2.4 Info	25
8.2.2.5 Model	26
8.2.2.6 Option	26
8.2.2.7 Source	27
8.2.2.8 Stream	28
8.3 Intrinsic & Extrinsic	29
8.3.1 详细描述	29
8.3.2 枚举类型说明	29
8.3.2.1 CalibrationModel	29
8.4 Datatypes	31
8.4.1 详细描述	31
9 类说明	33
9.1 mynteye::API 类参考	33
9.1.1 详细描述	35
9.1.2 成员类型定义说明	36
9.1.2.1 motion_callback_t	36
9.1.2.2 stream_callback_t	36
9.1.3 成员函数说明	36
9.1.3.1 Create() [1/3]	36
9.1.3.2 Create() [2/3]	37
9.1.3.3 Create() [3/3]	37
9.1.3.4 EnableStreamData()	38
9.1.3.5 GetIntrinsic()	38
9.1.3.6 GetStreamDatas()	38
9.2 mynteye::AsyncCallback< Data > 模板类参考	39
9.3 mynteye::Context 类参考	39
9.3.1 详细描述	39
9.3.2 成员函数说明	39
9.3.2.1 devices()	39
9.4 mynteye::Device 类参考	39
9.4.1 详细描述	42
9.4.2 成员类型定义说明	42
9.4.2.1 motion_callback_t	42

9.4.2.2 stream_callback_t	42
9.4.3 成员函数说明	43
9.4.3.1 Create()	43
9.4.3.2 GetLatestStreamData()	43
9.4.3.3 GetStreamDatas()	43
9.5 mynteye::DeviceInfo 结构体参考	44
9.5.1 详细描述	44
9.6 mynteye::Extrinsics 结构体参考	44
9.6.1 详细描述	44
9.6.2 成员函数说明	44
9.6.2.1 Inverse()	45
9.7 mynteye::device::Frame 类参考	45
9.7.1 详细描述	45
9.7.2 成员函数说明	46
9.7.2.1 clone()	46
9.7.2.2 data() [1/2]	46
9.7.2.3 data() [2/2]	46
9.7.2.4 format()	46
9.7.2.5 height()	46
9.7.2.6 size()	47
9.7.2.7 width()	47
9.8 mynteye::HardwareVersion 类参考	47
9.8.1 详细描述	47
9.9 mynteye::ImgData 结构体参考	47
9.9.1 详细描述	48
9.10 mynteye::device::ImgParams 结构体参考	48
9.11 mynteye::ImuData 结构体参考	48
9.11.1 详细描述	48
9.11.2 类成员变量说明	48
9.11.2.1 accel	48
9.11.2.2 flag	49
9.11.2.3 gyro	49
9.12 mynteye::ImuIntrinsics 结构体参考	49
9.12.1 详细描述	49

9.12.2 类成员变量说明	49
9.12.2.1 scale	50
9.13 mynteye::device::ImuParams 结构体参考	50
9.14 mynteye::IntrinsicsBase 结构体参考	50
9.15 mynteye::IntrinsicsEquidistant 结构体参考	50
9.15.1 详细描述	51
9.16 mynteye::IntrinsicsPinhole 结构体参考	51
9.16.1 详细描述	51
9.16.2 类成员变量说明	52
9.16.2.1 model	52
9.17 mynteye::device::MotionData 结构体参考	52
9.17.1 详细描述	52
9.17.2 类成员变量说明	52
9.17.2.1 imu	52
9.18 mynteye::api::MotionData 结构体参考	53
9.18.1 详细描述	53
9.18.2 类成员变量说明	53
9.18.2.1 imu	53
9.19 mynteye::MotionIntrinsics 结构体参考	53
9.19.1 详细描述	53
9.20 mynteye::Object 结构体参考	54
9.20.1 详细描述	54
9.21 mynteye::ObjMat 结构体参考	54
9.21.1 详细描述	55
9.22 mynteye::ObjMat2 结构体参考	55
9.22.1 详细描述	55
9.23 mynteye::OptionInfo 结构体参考	56
9.23.1 详细描述	56
9.24 mynteye::Plugin 类参考	56
9.24.1 详细描述	57
9.24.2 成员函数说明	57
9.24.2.1 OnCreate()	57
9.24.2.2 OnDepthProcess()	57
9.24.2.3 OnDisparityNormalizedProcess()	58

9.24.2.4 OnDisparityProcess()	58
9.24.2.5 OnPointsProcess()	58
9.24.2.6 OnRectifyProcess()	60
9.25 mynteye::Resolution 结构体参考	60
9.25.1 详细描述	61
9.26 mynteye::api::StreamData 结构体参考	61
9.26.1 详细描述	61
9.26.2 类成员变量说明	61
9.26.2.1 frame	61
9.26.2.2 frame_id	62
9.26.2.3 frame_raw	62
9.26.2.4 img	62
9.27 mynteye::device::StreamData 结构体参考	62
9.27.1 详细描述	62
9.27.2 类成员变量说明	63
9.27.2.1 frame	63
9.27.2.2 frame_id	63
9.27.2.3 img	63
9.28 mynteye::StreamRequest 结构体参考	63
9.28.1 详细描述	64
9.29 mynteye::strings_error 类参考	64
9.29.1 详细描述	64
9.30 mynteye::Type 类参考	64
9.30.1 详细描述	64
9.31 mynteye::Version 类参考	64
9.31.1 详细描述	64
索引	65

Chapter 1

MYNT EYE S SDK

- API 类
- API 模块
 - 枚举类型
 - 数据类型
 - 工具函数
 - 内参与外参
- 设备说明
 - 设备数据说明
 - 设备控制说明

Chapter 2

设备数据说明

- [硬件信息说明](#)
- [图像参数说明](#)
- [IMU 参数说明](#)
- [图像数据说明](#)
- [IMU 数据说明](#)

2.1 硬件信息说明

名称	字段	固定值	描述符获取	拓展通道获取	字节数	说明
VID	vid	0x04B4	√	×	2	
PID	pid	0x00F9	√	×	2	
设备名称	name	MYNT-E↔ YE-?	√	√ Get	16	MYNT-E↔ YE-S1000
序列号	serial_↔ number	-	√	√ Get	16	
固件版本	firmware↔ _version	-	√	√ Get	2	major,minor
硬件版本	hardware↔ _version	-	×	√ Get	3	major,minor,flag
协议版本	spec_↔ version	-	×	√ Get	2	major,minor
镜头类型	lens_type	-	×	√ Get/Set	4	vendor(2),product(2) , 未 Set 默 认 0

名称	字段	固定值	描述符获取	拓展通道获取	字节数	说明
IMU 类型	imu_type	-	×	√ Get/Set	4	vendor(2),product(2), 未 Set 默认 0
基线长度	nominal_ baseline	-	×	√ Get/Set	2	单位 mm , 未 set 默认 0
辅助芯片版本	auxiliary_ _chip_ version	-	×	√ Get	2	major,minor
ISP 版本	isp_version	-	×	√ Get	2	major,minor

- 描述符获取：指通用 USB 设备信息，可用工具查看。
- 拓展通道获取：指通过拓展通道（UVC Extension Unit）问硬件获取到的信息，需要读取。

2.2 图像参数说明

图像内参

名称	字段	单位	字节数	说明
宽度	width	px	2	uint16_t; [0,65535]
高度	height	px	2	uint16_t; [0,65535]
焦距	fx	-	8	double
	fy	-	8	double
图像中心	cx	-	8	double
	cy	-	8	double
畸变模型	model	-	1	uint8_t; pinhole,...
畸变参数	coeffs[5]	-	40	double; k1,k2,p1,p2,k3

图像分辨率不同，内参不同。多分辨率的话，需有多个内参。

图像外参

Left Image 到 Right Image 的变换矩阵。

名称	字段	单位	字节数	说明
旋转矩阵	rotation[3][3]	-	72	double
平移矩阵	translation[3]	-	24	double

2.3 IMU 参数说明

IMU 内参

名称	字段	单位	字节数	说明
比例因子	acc_scale[3][3]	-	72	double
	gyro_scale[3][3]	-	72	double
零漂	acc_drift[3]	-	24	double
	gyro_drift[3]	-	24	double
噪声密度	acc_noise[3]	-	24	double
	gyro_noise[3]	-	24	double
随机游走	acc_bias[3]	-	24	double
	gyro_bias[3]	-	24	double

IMU 外参

Left Image 到 IMU 的变换矩阵。

名称	字段	单位	字节数	说明
旋转矩阵	rotation[3][3]	-	72	double
平移矩阵	translation[3]	-	24	double

2.4 图像数据说明

名称	字段	单位	字节数	说明
帧 ID	frame_id	-	2	uint16_t; [0,65535]
时间戳	timestamp	1 us	8	uint64_t
曝光时间	exposure_time	1 us	2	uint16_t

图像数据传输方式：倒序排在图像尾部。

图像数据包

Name	Header	Size	FrameID	Timestamp	Exposure↔ Time	Checksum
字节数	1	1	2	8	2	1
类型	uint8↔ _t	uint8_t	uint16↔ _t	uint64_t	uint16_t	uint8_t

Name	Header	Size	FrameID	Timestamp	Exposure↔ Time	Checksum
描述	0x3B	0x10 (数据内容大小)	帧 ID	时间戳	曝光时间	校验码 (数据内容所有字节异或)

- 数据包校验不过，会丢弃该帧。
- 时间的单位精度为：1 us。
- 时间累计是从上电时从开始，而不是从打开时开始。

2.5 IMU 数据说明

IMU 请求数据包

Name	Header	Serial Number
字节数	1	4
类型	uint8↔ _t	uint32_t
描述	0x5A	首次请求写 0，不然写上次响应数据包最后一个 IMU 包的序列号

IMU 响应数据包

IMU 响应数据包里会包含 1 个 IMU 包，而每个 IMU 包又带有多个 IMU 段。

Name	Header	State	Size	IMU Packets	Checksum
字节数	1	1	2	...	1
类型	uint8↔ _t	uint8_t	uint16_t	-	uint8_t
描述	0x5B	正常状态为 0，否则错误	数据内容大小	所包含的 IMU 包	校验码 (数据内容所有字节异或)

IMU 包

IMU 包/小包，是一组 IMU 数据。

Name	Count	IMU Datas
字节数	2	...
类型	uint16_t	-
描述	IMU 段数量	所包含的 IMU 段

IMU 段

Name	Serial Number	Timestamp	flag	Temperature	Accelerometer or Gyroscope
字节数	4	8	1	2	6
类型	uint32_t	uint64_t	int8_t	int16_t	int16_t * 3
Description	序列号	时间戳	指定传感器类型	IMU 的温度	陀螺仪或陀螺仪 x y z 三轴的值

- 加速度计和陀螺仪的计量值换算成物理值公式： $real = data * range / 0x10000$ 。
 - 加速度计量程默认值为 12 g，陀螺仪量程默认值为 1000 deg/s。
- 温度计量值换算成物理值公式： $real = data / ratio + offset$ 。
 - ratio 默认值为 326.8，offset 默认值为 25°C。

Chapter 3

设备控制说明

- [控制 API 说明](#)
- [拓展通道说明](#)

3.1 控制 API 说明

控制有两种实现方式，一是通过 UVC 标准协议，二是通过 UVC 拓展通道自定义协议。

标准协议

名称	字段	字节数	默认值	最小值	最大值	是否储存	Flash 地址	说明
亮度	brightness	2	192	0	255	√	0x14	关闭自动曝光，手动设定的参数

UVC 标准协议实现的控制，有现成的 API 进行 Get & Set ，包括 Min, Max, Default 。

自定义协议

名称	字段	字节数	默认值	最小值	最大值	是否储存	Flash地址	所属通道	通道地址	说明
曝光模式	exposure← _← mode	1	0	0	1	√	0x0F	XU← _C← AM← _C← TRL	0x0100	0: 开启自动曝光; 1: 关闭
最大增益	max← _gain	2	8	0	255	√	0x1D	XU← _C← AM← _C← TRL	0x0100	开始自动曝光, 可设置的阈值
最大曝光时间	max← _← exposure← _time	2	333	0	1000	√	0x1B	XU← _C← AM← _C← TRL	0x0100	开始自动曝光, 可设置的阈值
期望亮度	desired← _← brightness	2	122	1	255	√	0x19	XU← _C← AM← _C← TRL	0x0100	
擦除芯片	erase← _chip		-	-	-	×	-	XU← _H← AL← F_← DU← PLEX	0x0200	
最小曝光时间	min← _← exposure← _time	2	0	0	1000	√	-	XU← _C← AM← _C← TRL	0x0100	开始自动曝光, 可设置的阈值
加速度计量程	accelerometer← _← range	2	12	6	48	√	-	XU← _C← AM← _C← TRL	0x0100	

名称	字段	字节数	默认值	最小值	最大值	是否储存	Flash地址	所属通道	通道地址	说明
陀螺仪量程	gyroscope_range	2	1000	250	4000	√	-	XU_C_AM_C_TRL	0x0100	
加速度计低通滤波	accelerometer_low_pass_filter	2	2	0	2	√	-	XU_C_AM_C_TRL	0x0100	
陀螺仪低通滤波	gyroscope_low_pass_filter	2	64	23	64	√	-	XU_C_AM_C_TRL	0x0100	

3.2 拓展通道说明

名称	字段	地址	带宽	说明
相机控制通道	XU_CAM_CTRL_CHANNEL	1	3	
半双工通道	XU_HALF_DUPLEX_CHANNEL	2	20	
IMU 请求通道	XU_IMUDATA_WRITE_CHANNEL	3	5	
IMU 响应通道	XU_IMUDATA_READ_CHANNEL	4	2000	
文件通道	XU_FILE_CHANNEL	5	2000	

相机控制通道

相机控制通道是那些需要 Get & Set & Query 的控制通道，其中 Query 细分为 Min, Max, Default。

半双工通道

半双工通道是那些仅需 Set 的控制通道，如请求零漂矫正。

IMU 通道

用来请求和响应 IMU 数据的通道，可参见 [IMU 数据说明](#)。

文件通道

用来读写硬件信息、图像参数、IMU 参数的通道。

Name	Header	Size	File	Checksum
字节数	1	2	-	1
类型	uint8↔ _t	uint16_t	-	uint8_t
描述	标识	文件大小	文件内容	校验码（文件内容所有字节异或）

Header Bit Subscript	Description
0	硬件信息
1	图像参数
2	IMU 参数
3~6	未定义
7	0: Get; 1: Set

文件内容包

Name	ID	Size	Content
字节数	1	2	-
类型	uint8↔ _t	uint16↔ _t	-
描述	内容 ID	内容大小	内容

File	ID	Max Size
硬件信息	1	250
图像参数	2	404
IMU 参数	4	500

Chapter 4

弃用列表

成员 [mynteye::API::GetIntrinsics](#) (const Stream &stream) const

Get the intrinsics (pinhole) of stream.

成员 [mynteye::Device::GetLatestStreamData](#) (const Stream &stream)

Replaced by [GetStreamData\(const Stream &stream\)](#)

成员 [mynteye::IntrinsicsPinhole::model](#)

Replaced by `calib_model_`.

Chapter 5

模块索引

5.1 模块

这里列出了所有模块:

Utilities	21
Enumerations	23
Intrinsics & Extrinsics	29
Datatypes	31

Chapter 6

继承关系索引

6.1 类继承关系

此继承关系列表按字典顺序粗略的排序:

mynteye::API	33
mynteye::AsyncCallback< Data >	39
mynteye::Context	39
mynteye::Device	39
mynteye::DeviceInfo	44
mynteye::Extrinsics	44
mynteye::device::Frame	45
mynteye::ImgData	47
mynteye::device::ImgParams	48
mynteye::ImuData	48
mynteye::ImuIntrinsics	49
mynteye::device::ImuParams	50
mynteye::IntrinsicsBase	50
mynteye::IntrinsicsEquidistant	50
mynteye::IntrinsicsPinhole	51
mynteye::device::MotionData	52
mynteye::api::MotionData	53
mynteye::MotionIntrinsics	53
mynteye::Object	54
mynteye::ObjMat	54
mynteye::ObjMat2	55
mynteye::OptionInfo	56
mynteye::Plugin	56

mynteye::Resolution	60
runtime_error	
mynteye::strings_error	64
mynteye::api::StreamData	61
mynteye::device::StreamData	62
mynteye::StreamRequest	63
mynteye::Type	64
mynteye::Version	64
mynteye::HardwareVersion	47

Chapter 7

类索引

7.1 类列表

这里列出了所有类、结构、联合以及接口定义等，并附带简要说明:

mynteye::API	
To communicate with MYNT® EYE device	33
mynteye::AsyncCallback< Data >	39
mynteye::Context	
The context about devices	39
mynteye::Device	
To communicate with MYNT® EYE device	39
mynteye::DeviceInfo	
Device infomation	44
mynteye::Extrinsics	
Extrinsics, represent how the different datas are connected	44
mynteye::device::Frame	
Frame with raw data	45
mynteye::HardwareVersion	
Hardware version	47
mynteye::ImgData	
Image data	47
mynteye::device::ImgParams	48
mynteye::ImuData	
IMU data	48
mynteye::ImuIntrinsics	
IMU intrinsics: scale, drift and variances	49
mynteye::device::ImuParams	50
mynteye::IntrinsicsBase	50

mynteye::IntrinsicsEquidistant	
Stream intrinsics (Equidistant: KANNALA_BRANDT)	50
mynteye::IntrinsicsPinhole	
Stream intrinsics (Pinhole)	51
mynteye::device::MotionData	
Device motion data	52
mynteye::api::MotionData	
API motion data	53
mynteye::MotionIntrinsics	
Motion intrinsics, including accelerometer and gyroscope	53
mynteye::Object	
Input & output object	54
mynteye::ObjMat	
Input & output object of one cv::Mat	54
mynteye::ObjMat2	
Input & output object of two cv::Mat	55
mynteye::OptionInfo	
Option info	56
mynteye::Plugin	
The plugin which could implement processing by yourself	56
mynteye::Resolution	
Resolution	60
mynteye::api::StreamData	
API stream data	61
mynteye::device::StreamData	
Device stream data	62
mynteye::StreamRequest	
Stream request	63
mynteye::strings_error	
The strings error	64
mynteye::Type	
Type	64
mynteye::Version	
Version	64

Chapter 8

模块说明

8.1 Utilities

函数

- MYNTEYE_API std::shared_ptr< Device > mynteye::device::select ()
Detecting MYNTEYE devices and prompt user to select one.
- MYNTEYE_API MYNTEYE_NAMESPACE::StreamRequest mynteye::device::select_request (const std::shared_ptr< Device > &device, bool *ok)
List stream requests and prompt user to select one.
- MYNTEYE_API float mynteye::utils::get_real_exposure_time (std::int32_t frame_rate, std::uint16_t exposure_time)
Get real exposure time in ms from virtual value, according to its frame rate.
- MYNTEYE_API std::string mynteye::utils::get_sdk_root_dir ()
Get sdk root dir.
- MYNTEYE_API std::string mynteye::utils::get_sdk_install_dir ()
Get sdk install dir.

8.1.1 详细描述

8.1.2 函数说明

8.1.2.1 get_real_exposure_time()

```
MYNTEYE_API float mynteye::utils::get_real_exposure_time (  
    std::int32_t frame_rate,  
    std::uint16_t exposure_time )
```

Get real exposure time in ms from virtual value, according to its frame rate.

参数

frame_rate	the frame rate of the device.
exposure_time	the virtual exposure time.

返回

the real exposure time in ms, or the virtual value if frame rate is invalid.

8.1.2.2 select()

MYNTEYE_API std::shared_ptr<Device> mynteye::device::select ()

Detecting MYNT EYE devices and prompt user to select one.

返回

the selected device, or nullptr if none.

8.1.2.3 select_request()

```
MYNTEYE_API MYNTEYE_NAMESPACE::StreamRequest mynteye::device::select_request (
    const std::shared_ptr< Device > & device,
    bool * ok )
```

List stream requests and prompt user to select one.

返回

the selected request.

8.2 Enumerations

Public enumeration types.

枚举

- enum `mynteye::Model` : `std::uint8_t` { `mynteye::Model::STANDARD`, `mynteye::Model::STANDARD2`, `mynteye::Model::STANDARD210A`, `mynteye::Model::LAST` }
Device model.
- enum `mynteye::Stream` : `std::uint8_t` {
`mynteye::Stream::LEFT`, `mynteye::Stream::RIGHT`, `mynteye::Stream::LEFT_RECTIFIED`,
`mynteye::Stream::RIGHT_RECTIFIED`,
`mynteye::Stream::DISPARITY`, `mynteye::Stream::DISPARITY_NORMALIZED`, `mynteye::Stream::DEPTH`,
`mynteye::Stream::POINTS`,
`mynteye::Stream::LAST` }
Streams define different type of data.
- enum `mynteye::Capabilities` : `std::uint8_t` {
`mynteye::Capabilities::STEREO`, `mynteye::Capabilities::STEREO_COLOR`, `mynteye::Capabilities::COLOR`,
`mynteye::Capabilities::DEPTH`,
`mynteye::Capabilities::POINTS`, `mynteye::Capabilities::FISHEYE`, `mynteye::Capabilities::INFRARED`,
`mynteye::Capabilities::INFRARED2`,
`mynteye::Capabilities::IMU`, `mynteye::Capabilities::LAST` }
Capabilities define the full set of functionality that the device might provide.
- enum `mynteye::Info` : `std::uint8_t` {
`mynteye::Info::DEVICE_NAME`, `mynteye::Info::SERIAL_NUMBER`, `mynteye::Info::FIRMWARE_VERSION`,
`mynteye::Info::HARDWARE_VERSION`,
`mynteye::Info::SPEC_VERSION`, `mynteye::Info::LENS_TYPE`, `mynteye::Info::IMU_TYPE`,
`mynteye::Info::NOMINAL_BASELINE`,
`mynteye::Info::LAST` }
Camera info fields are read-only strings that can be queried from the device.
- enum `mynteye::Option` : `std::uint8_t` {
`mynteye::Option::GAIN`, `mynteye::Option::BRIGHTNESS`, `mynteye::Option::CONTRAST`,
`mynteye::Option::FRAME_RATE`,
`mynteye::Option::IMU_FREQUENCY`, `mynteye::Option::EXPOSURE_MODE`, `mynteye::Option::MAX_GAIN`,
`mynteye::Option::MAX_EXPOSURE_TIME`,
`mynteye::Option::MIN_EXPOSURE_TIME`, `mynteye::Option::DESIRED_BRIGHTNESS`,
`mynteye::Option::IR_CONTROL`, `mynteye::Option::HDR_MODE`,
`mynteye::Option::ACCELEROMETER_RANGE`, `mynteye::Option::GYROSCOPE_RANGE`,
`mynteye::Option::ACCELEROMETER_LOW_PASS_FILTER`, `mynteye::Option::GYROSCOPE_LOW_PASS_FILTER`,
`mynteye::Option::ZERO_DRIFT_CALIBRATION`, `mynteye::Option::ERASE_CHIP`,
`mynteye::Option::LAST` }

Camera control options define general configuration controls.

- enum `mynteye::Source` : `std::uint8_t` { `mynteye::Source::VIDEO_STREAMING`, `mynteye::Source::MOTION`, `mynteye::Source::ALL`, `mynteye::Source::LAST` }

Source allows the user to choose which data to be captured.

- enum `mynteye::AddOns` : `std::uint8_t` { `mynteye::AddOns::INFRARED`, `mynteye::AddOns::INFRARED2`, `mynteye::AddOns::LAST` }

Add-Ons are peripheral modules of our hardware.

- enum `mynteye::Format` : `std::uint32_t` { `mynteye::Format::GREY` = `((std::uint32_t)('G') | ((std::uint32_t)('R') << 8) | ((std::uint32_t)('E') << 16) | ((std::uint32_t)('Y') << 24))`, `mynteye::Format::YUYV` = `((std::uint32_t)('Y') | ((std::uint32_t)('U') << 8) | ((std::uint32_t)('Y') << 16) | ((std::uint32_t)('V') << 24))`, `mynteye::Format::BGR888` = `((std::uint32_t)('B') | ((std::uint32_t)('G') << 8) | ((std::uint32_t)('R') << 16) | ((std::uint32_t)('3') << 24))`, `mynteye::Format::RGB888` = `((std::uint32_t)('R') | ((std::uint32_t)('G') << 8) | ((std::uint32_t)('B') << 16) | ((std::uint32_t)('3') << 24))`, `mynteye::Format::LAST` }

Formats define how each stream can be encoded.

8.2.1 详细描述

Public enumeration types.

8.2.2 枚举类型说明

8.2.2.1 AddOns

```
enum mynteye::AddOns : std::uint8_t [strong]
```

Add-Ons are peripheral modules of our hardware.

枚举值

INFRARED	Infrared
INFRARED2	Second infrared
LAST	Last guard

8.2.2.2 Capabilities

enum `mynteye::Capabilities` : `std::uint8_t` [strong]

Capabilities define the full set of functionality that the device might provide.

枚举值

STEREO	Provides stereo stream
STEREO_COLOR	Provide stereo color stream
COLOR	Provides color stream
DEPTH	Provides depth stream
POINTS	Provides point cloud stream
FISHEYE	Provides fisheye stream
INFRARED	Provides infrared stream
INFRARED2	Provides second infrared stream
IMU	Provides IMU (accelerometer, gyroscope) data
LAST	Last guard

8.2.2.3 Format

enum `mynteye::Format` : `std::uint32_t` [strong]

Formats define how each stream can be encoded.

枚举值

GREY	Greyscale, 8 bits per pixel
YUYV	YUV 4:2:2, 16 bits per pixel
BGR888	BGR 8:8:8, 24 bits per pixel
RGB888	RGB 8:8:8, 24 bits per pixel
LAST	Last guard

8.2.2.4 Info

enum `mynteye::Info` : `std::uint8_t` [strong]

Camera info fields are read-only strings that can be queried from the device.

枚举值

DEVICE_NAME	Device name
SERIAL_NUMBER	Serial number
FIRMWARE_VERSION	Firmware version
HARDWARE_VERSION	Hardware version
SPEC_VERSION	Spec version
LENS_TYPE	Lens type
IMU_TYPE	IMU type
NOMINAL_BASELINE	Nominal baseline
LAST	Last guard

8.2.2.5 Model

```
enum mynteye::Model : std::uint8_t [strong]
```

Device model.

枚举值

STANDARD	Standard
STANDARD2	Standard 2
STANDARD210A	Standard 210a
LAST	Last guard

8.2.2.6 Option

```
enum mynteye::Option : std::uint8_t [strong]
```

Camera control options define general configuration controls.

枚举值

GAIN	Image gain, valid if manual-exposure range: [0,48], default: 24
BRIGHTNESS	Image brightness, valid if manual-exposure range: [0,240], default: 120
CONTRAST	Image contrast, valid if manual-exposure range: [0,255], default: 127

枚举值

FRAME_RATE	Image frame rate, must set IMU_FREQUENCY together values: {10,15,20,25,30,35,40,45,50,55,60}, default: 25
IMU_FREQUENCY	IMU frequency, must set FRAME_RATE together values: {100,200,250,333,500}, default: 200
EXPOSURE_MODE	Exposure mode 0: enable auto-exposure 1: disable auto-exposure (manual-exposure)
MAX_GAIN	Max gain, valid if auto-exposure range of standard 1: [0,48], default: 48 range of standard 2: [0,255], default: 8
MAX_EXPOSURE_TIME	Max exposure time, valid if auto-exposure range of standard 1: [0,240], default: 240 range of standard 2: [0,1000], default: 333
MIN_EXPOSURE_TIME	min exposure time, valid if auto-exposure range: [0,1000], default: 0
DESIRED_BRIGHTNESS	Desired brightness, valid if auto-exposure range of standard 1: [0,255], default: 192 range of standard 2: [1,255], default: 122
IR_CONTROL	IR control range: [0,160], default: 0
HDR_MODE	HDR mode 0: 10-bit 1: 12-bit
ACCELEROMETER_RANGE	The range of accelerometer value of standard 1: {4,8,16,32}, default: 8 value of standard 2: {6,12,24,48}, default: 12
GYROSCOPE_RANGE	The range of gyroscope value of standard 1: {500,1000,2000,4000}, default: 1000 value of standard 2: {250,500,1000,2000,4000}, default: 1000
ACCELEROMETER_LOW_PASS_FILTER	The parameter of accelerometer low pass filter values: {0,1,2}, default: 2
GYROSCOPE_LOW_PASS_FILTER	The parameter of gyroscope low pass filter values: {23,64}, default: 64
ZERO_DRIFT_CALIBRATION	Zero drift calibration
ERASE_CHIP	Erase chip
LAST	Last guard

8.2.2.7 Source

```
enum mynteye::Source : std::uint8_t [strong]
```

Source allows the user to choose which data to be captured.

枚举值

VIDEO_STREAMING	Video streaming of stereo, color, depth, etc.
MOTION_TRACKING	Motion tracking of IMU (accelerometer, gyroscope)
ALL	Enable everything together
LAST	Last guard

8.2.2.8 Stream

```
enum mynteye::Stream : std::uint8_t [strong]
```

Streams define different type of data.

枚举值

LEFT	Left stream
RIGHT	Right stream
LEFT_RECTIFIED	Left stream, rectified
RIGHT_RECTIFIED	Right stream, rectified
DISPARITY	Disparity stream
DISPARITY_NORMALIZED	Disparity stream, normalized
DEPTH	Depth stream
POINTS	Point cloud stream
LAST	Last guard

8.3 Intrinsic & Extrinsic

Intrinsic and extrinsic properties.

类

- struct `mynteye::IntrinsicPinhole`
Stream intrinsic (Pinhole)
- struct `mynteye::IntrinsicEquidistant`
Stream intrinsic (Equidistant: KANNALA_BRANDT)
- struct `mynteye::ImuIntrinsic`
IMU intrinsic: scale, drift and variances.
- struct `mynteye::MotionIntrinsic`
Motion intrinsic, including accelerometer and gyroscope.
- struct `mynteye::Extrinsic`
`Extrinsic`, represent how the different datas are connected.

枚举

- enum `mynteye::CalibrationModel` : `std::uint8_t` { `mynteye::CalibrationModel::PINHOLE` = 0, `mynteye::CalibrationModel::KANNALA_BRANDT` = 1, `mynteye::CalibrationModel::UNKNOWN` }
- Camera calibration model.

8.3.1 详细描述

Intrinsic and extrinsic properties.

8.3.2 枚举类型说明

8.3.2.1 CalibrationModel

```
enum mynteye::CalibrationModel : std::uint8_t [strong]
```

Camera calibration model.

枚举值

PINHOLE	Pinhole
KANNALA_BRANDT	Equidistant: KANNALA_BRANDT
UNKNOW	Unknow

8.4 Datatypes

Public data types.

类

- struct `mynteye::api::StreamData`
API stream data.
- struct `mynteye::api::MotionData`
API motion data.
- class `mynteye::device::Frame`
Frame with raw data.
- struct `mynteye::device::StreamData`
Device stream data.
- struct `mynteye::device::MotionData`
Device motion data.
- struct `mynteye::DeviceInfo`
Device information.
- struct `mynteye::ImgData`
Image data.
- struct `mynteye::ImuData`
IMU data.
- struct `mynteye::OptionInfo`
Option info.

8.4.1 详细描述

Public data types.

Chapter 9

类说明

9.1 mynteye::API 类参考

The [API](#) class to communicate with MYNT® EYE device.

Public 类型

- using [stream_callback_t](#) = std::function< void(const [api::StreamData](#) &data)>
The [api::StreamData](#) callback.
- using [motion_callback_t](#) = std::function< void(const [api::MotionData](#) &data)>
The [api::MotionData](#) callback.

Public 成员函数

- [Model GetModel](#) () const
Get the model.
- bool [Supports](#) (const [Stream](#) &stream) const
Supports the stream or not.
- bool [Supports](#) (const [Capabilities](#) &capability) const
Supports the capability or not.
- bool [Supports](#) (const [Option](#) &option) const
Supports the option or not.
- bool [Supports](#) (const [AddOns](#) &addon) const
Supports the addon or not.
- [StreamRequest SelectStreamRequest](#) (bool *ok) const
Log all stream requests and prompt user to select one.

- `const std::vector< StreamRequest > & GetStreamRequests (const Capabilities &capability) const`
Get all stream requests of the capability.
- `void ConfigStreamRequest (const Capabilities &capability, const StreamRequest &request)`
Config the stream request to the capability.
- `const StreamRequest & GetStreamRequest (const Capabilities &capability) const`
Get the config stream requests of the capability.
- `const std::vector< StreamRequest > & GetStreamRequests () const`
Get all stream requests of the key stream capability.
- `void ConfigStreamRequest (const StreamRequest &request)`
Config the stream request to the key stream capability.
- `const StreamRequest & GetStreamRequest () const`
Get the config stream requests of the key stream capability.
- `std::shared_ptr< DeviceInfo > GetInfo () const`
Get the device info.
- `std::string GetInfo (const Info &info) const`
Get the device info.
- `IntrinsicsPinhole GetIntrinsics (const Stream &stream) const`
- `template<typename T >`
`T GetIntrinsics (const Stream &stream) const`
Get the intrinsics of stream.
- `std::shared_ptr< IntrinsicsBase > GetIntrinsicsBase (const Stream &stream) const`
Get the intrinsics base of stream.
- `Extrinsics GetExtrinsics (const Stream &from, const Stream &to) const`
Get the extrinsics from one stream to another.
- `MotionIntrinsics GetMotionIntrinsics () const`
Get the intrinsics of motion.
- `Extrinsics GetMotionExtrinsics (const Stream &from) const`
Get the extrinsics from one stream to motion.
- `void LogOptionInfos () const`
Log all option infos.
- `OptionInfo GetOptionInfo (const Option &option) const`
Get the option info.
- `std::int32_t GetOptionValue (const Option &option) const`
Get the option value.
- `void SetOptionValue (const Option &option, std::int32_t value)`
Set the option value.
- `bool RunOptionAction (const Option &option) const`
Run the option action.
- `void SetStreamCallback (const Stream &stream, stream_callback_t callback)`

- Set the callback of stream.

 - void [SetMotionCallback](#) ([motion_callback_t](#) callback)

Set the callback of motion.
- bool [HasStreamCallback](#) (const [Stream](#) &stream) const

Has the callback of stream.
- bool [HasMotionCallback](#) () const

Has the callback of motion.
- void [Start](#) (const [Source](#) &source)

Start capturing the source.
- void [Stop](#) (const [Source](#) &source)

Stop capturing the source.
- void [WaitForStreams](#) ()

Wait the streams are ready.
- void [EnableStreamData](#) (const [Stream](#) &stream)

Enable the data of stream.
- void [DisableStreamData](#) (const [Stream](#) &stream)

Disable the data of stream.
- [api::StreamData](#) [GetStreamData](#) (const [Stream](#) &stream)

Get the latest data of stream.
- std::vector< [api::StreamData](#) > [GetStreamDatas](#) (const [Stream](#) &stream)

Get the datas of stream.
- void [EnableMotionDatas](#) (std::size_t max_size=std::numeric_limits< std::size_t >::max())

Enable cache motion datas.
- std::vector< [api::MotionData](#) > [GetMotionDatas](#) ()

Get the motion datas.
- void [EnablePlugin](#) (const std::string &path)

Enable the plugin.

静态 Public 成员函数

- static std::shared_ptr< [API](#) > [Create](#) (int argc, char *argv[])

Create the [API](#) instance.
- static std::shared_ptr< [API](#) > [Create](#) (int argc, char *argv[], const std::shared_ptr< [Device](#) > &device)

Create the [API](#) instance.
- static std::shared_ptr< [API](#) > [Create](#) (const std::shared_ptr< [Device](#) > &device)

Create the [API](#) instance.

9.1.1 详细描述

The [API](#) class to communicate with MYNT® EYE device.

9.1.2 成员类型定义说明

9.1.2.1 motion_callback_t

```
using mynteye::API::motion_callback_t = std::function<void(const api::MotionData &data)>
```

The [api::MotionData](#) callback.

9.1.2.2 stream_callback_t

```
using mynteye::API::stream_callback_t = std::function<void(const api::StreamData &data)>
```

The [api::StreamData](#) callback.

9.1.3 成员函数说明

9.1.3.1 Create() [1/3]

```
static std::shared_ptr<API> mynteye::API::Create (
    int argc,
    char * argv[] ) [static]
```

Create the [API](#) instance.

参数

argc	the arg count.
argv	the arg values.

返回

the [API](#) instance.

注解

This will init glog with args and call `device::select()` to select a device.

9.1.3.2 Create() [2/3]

```
static std::shared_ptr<API> mynteye::API::Create (  
    int argc,  
    char * argv[],  
    const std::shared_ptr< Device > & device ) [static]
```

Create the [API](#) instance.

参数

argc	the arg count.
argv	the arg values.
device	the selected device.

返回

the [API](#) instance.

注解

This will init glog with args.

9.1.3.3 Create() [3/3]

```
static std::shared_ptr<API> mynteye::API::Create (  
    const std::shared_ptr< Device > & device ) [static]
```

Create the [API](#) instance.

参数

device	the selected device.
--------	----------------------

返回

the [API](#) instance.

9.1.3.4 EnableStreamData()

```
void mynteye::API::EnableStreamData (
    const Stream & stream )
```

Enable the data of stream.

注解

must enable the stream if it's a synthetic one. This means the stream is not native, the device has the capability to provide this stream, but still support this stream.

9.1.3.5 GetIntrinsics()

```
IntrinsicsPinhole mynteye::API::GetIntrinsics (
    const Stream & stream ) const
```

弃用 Get the intrinsics (pinhole) of stream.

9.1.3.6 GetStreamDatas()

```
std::vector<api::StreamData> mynteye::API::GetStreamDatas (
    const Stream & stream )
```

Get the datas of stream.

注解

default cache 4 datas at most.

9.2 mynteye::AsyncCallback< Data > 模板类参考

9.3 mynteye::Context 类参考

The context about devices.

Public 成员函数

- `std::vector< std::shared_ptr< Device > > devices () const`
Get all devices now.

9.3.1 详细描述

The context about devices.

9.3.2 成员函数说明

9.3.2.1 devices()

`std::vector<std::shared_ptr<Device> > mynteye::Context::devices () const` [inline]

Get all devices now.

返回

a vector of all devices.

9.4 mynteye::Device 类参考

The `Device` class to communicate with MYNT® EYE device.

Public 类型

- using `stream_callback_t` = `device::StreamCallback`
The `device::StreamData` callback.
- using `motion_callback_t` = `device::MotionCallback`
The `device::MotionData` callback.

Public 成员函数

- [Model GetModel](#) () const
Get the model.
- bool [Supports](#) (const [Stream](#) &stream) const
Supports the stream or not.
- bool [Supports](#) (const [Capabilities](#) &capability) const
Supports the capability or not.
- bool [Supports](#) (const [Option](#) &option) const
Supports the option or not.
- bool [Supports](#) (const [AddOns](#) &addon) const
Supports the addon or not.
- const std::vector< [StreamRequest](#) > & [GetStreamRequests](#) (const [Capabilities](#) &capability) const
Get all stream requests of the capability.
- void [ConfigStreamRequest](#) (const [Capabilities](#) &capability, const [StreamRequest](#) &request)
Config the stream request to the capability.
- const [StreamRequest](#) & [GetStreamRequest](#) (const [Capabilities](#) &capability) const
Get the config stream requests of the capability.
- const std::vector< [StreamRequest](#) > & [GetStreamRequests](#) () const
Get all stream requests of the key stream capability.
- void [ConfigStreamRequest](#) (const [StreamRequest](#) &request)
Config the stream request to the key stream capability.
- const [StreamRequest](#) & [GetStreamRequest](#) () const
Get the config stream requests of the key stream capability.
- std::shared_ptr< [DeviceInfo](#) > [GetInfo](#) () const
Get the device info.
- std::string [GetInfo](#) (const [Info](#) &info) const
Get the device info of a field.
- std::shared_ptr< [IntrinsicsBase](#) > [GetIntrinsics](#) (const [Stream](#) &stream) const
Get the intrinsics of stream.
- [Extrinsics GetExtrinsics](#) (const [Stream](#) &from, const [Stream](#) &to) const
Get the extrinsics from one stream to another.
- [MotionIntrinsics GetMotionIntrinsics](#) () const
Get the intrinsics of motion.
- [Extrinsics GetMotionExtrinsics](#) (const [Stream](#) &from) const
Get the extrinsics from one stream to motion.
- std::shared_ptr< [IntrinsicsBase](#) > [GetIntrinsics](#) (const [Stream](#) &stream, bool *ok) const
Get the intrinsics of stream.
- [Extrinsics GetExtrinsics](#) (const [Stream](#) &from, const [Stream](#) &to, bool *ok) const

- Get the extrinsics from one stream to another.

 - [MotionIntrinsics GetMotionIntrinsics](#) (bool *ok) const

Get the intrinsics of motion.
- [Extrinsics GetMotionExtrinsics](#) (const [Stream](#) &from, bool *ok) const

Get the extrinsics from one stream to motion.
- void [SetIntrinsics](#) (const [Stream](#) &stream, const std::shared_ptr< [IntrinsicsBase](#) > &in)

Set the intrinsics of stream.
- void [SetExtrinsics](#) (const [Stream](#) &from, const [Stream](#) &to, const [Extrinsics](#) &ex)

Set the extrinsics from one stream to another.
- void [SetMotionIntrinsics](#) (const [MotionIntrinsics](#) &in)

Set the intrinsics of motion.
- void [SetMotionExtrinsics](#) (const [Stream](#) &from, const [Extrinsics](#) &ex)

Set the extrinsics from one stream to motion.
- void [LogOptionInfos](#) () const

Log all option infos.
- [OptionInfo GetOptionInfo](#) (const [Option](#) &option) const

Get the option info.
- std::int32_t [GetOptionValue](#) (const [Option](#) &option) const

Get the option value.
- void [SetOptionValue](#) (const [Option](#) &option, std::int32_t value)

Set the option value.
- bool [RunOptionAction](#) (const [Option](#) &option) const

Run the option action.
- void [SetStreamCallback](#) (const [Stream](#) &stream, [stream_callback_t](#) callback, bool async=false)

Set the callback of stream.
- void [SetMotionCallback](#) ([motion_callback_t](#) callback, bool async=false)

Set the callback of motion.
- bool [HasStreamCallback](#) (const [Stream](#) &stream) const

Has the callback of stream.
- bool [HasMotionCallback](#) () const

Has the callback of motion.
- virtual void [Start](#) (const [Source](#) &source)

Start capturing the source.
- virtual void [Stop](#) (const [Source](#) &source)

Stop capturing the source.
- void [WaitForStreams](#) ()

Wait the streams are ready.
- [device::StreamData GetStreamData](#) (const [Stream](#) &stream)

Get the latest data of stream.

- `device::StreamData GetLatestStreamData (const Stream &stream)`
- `std::vector< device::StreamData > GetStreamDatas (const Stream &stream)`
Get the datas of stream.
- `void EnableMotionDatas ()`
Enable cache motion datas.
- `void EnableMotionDatas (std::size_t max_size)`
Enable cache motion datas.
- `std::vector< device::MotionData > GetMotionDatas ()`
Get the motion datas.

静态 Public 成员函数

- `static std::shared_ptr< Device > Create (const std::string &name, std::shared_ptr< uvc->::device > device)`
Create the `Device` instance.

9.4.1 详细描述

The `Device` class to communicate with MYNT® EYE device.

9.4.2 成员类型定义说明

9.4.2.1 motion_callback_t

```
using mynteye::Device::motion_callback_t = device::MotionCallback
```

The `device::MotionData` callback.

9.4.2.2 stream_callback_t

```
using mynteye::Device::stream_callback_t = device::StreamCallback
```

The `device::StreamData` callback.

9.4.3 成员函数说明

9.4.3.1 Create()

```
static std::shared_ptr<Device> mynteye::Device::Create (
    const std::string & name,
    std::shared_ptr< uvc::device > device ) [static]
```

Create the [Device](#) instance.

参数

name	the device name.
device	the device from uvc.

返回

the [Device](#) instance.

9.4.3.2 GetLatestStreamData()

```
device::StreamData mynteye::Device::GetLatestStreamData (
    const Stream & stream )
```

弃用 Replaced by [GetStreamData\(const Stream &stream\)](#)

9.4.3.3 GetStreamDatas()

```
std::vector<device::StreamData> mynteye::Device::GetStreamDatas (
    const Stream & stream )
```

Get the datas of stream.

注解

default cache 4 datas at most.

9.5 mynteye::DeviceInfo 结构体参考

[Device](#) infomation.

9.5.1 详细描述

[Device](#) infomation.

9.6 mynteye::Extrinsics 结构体参考

[Extrinsics](#), represent how the different datas are connected.

Public 成员函数

- [Extrinsics Inverse](#) () const
Inverse this extrinsics.

Public 属性

- double [rotation](#) [3][3]
Rotation matrix
- double [translation](#) [3]
Translation vector

9.6.1 详细描述

[Extrinsics](#), represent how the different datas are connected.

9.6.2 成员函数说明

9.6.2.1 Inverse()

[Extrinsics](#) mynteye::Extrinsics::Inverse () const [inline]

Inverse this extrinsics.

返回

the inversed extrinsics.

9.7 mynteye::device::Frame 类参考

[Frame](#) with raw data.

Public 成员函数

- [Frame](#) (const [StreamRequest](#) &request, const void *data)
Construct the frame with [StreamRequest](#) and raw data.
- [Frame](#) (std::uint16_t width, std::uint16_t height, [Format format](#), const void *data)
Construct the frame with stream info and raw data.
- std::uint16_t [width](#) () const
Get the width.
- std::uint16_t [height](#) () const
Get the height.
- [Format format](#) () const
Get the format.
- std::uint8_t * [data](#) ()
Get the data.
- const std::uint8_t * [data](#) () const
Get the const data.
- std::size_t [size](#) () const
Get the size of data.
- [Frame clone](#) () const
Clone a new frame.

9.7.1 详细描述

[Frame](#) with raw data.

9.7.2 成员函数说明

9.7.2.1 clone()

`Frame mynteye::device::Frame::clone () const` [inline]

Clone a new frame.

9.7.2.2 data() [1/2]

`std::uint8_t* mynteye::device::Frame::data ()` [inline]

Get the data.

9.7.2.3 data() [2/2]

`const std::uint8_t* mynteye::device::Frame::data () const` [inline]

Get the const data.

9.7.2.4 format()

`Format mynteye::device::Frame::format () const` [inline]

Get the format.

9.7.2.5 height()

`std::uint16_t mynteye::device::Frame::height () const` [inline]

Get the height.

9.7.2.6 size()

```
std::size_t mynteye::device::Frame::size ( ) const [inline]
```

Get the size of data.

9.7.2.7 width()

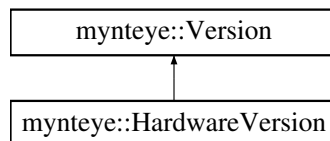
```
std::uint16_t mynteye::device::Frame::width ( ) const [inline]
```

Get the width.

9.8 mynteye::HardwareVersion 类参考

Hardware version.

类 mynteye::HardwareVersion 继承关系图:



9.8.1 详细描述

Hardware version.

9.9 mynteye::ImgData 结构体参考

Image data.

Public 属性

- `std::uint16_t frame_id`
Image frame id
- `std::uint64_t timestamp`
Image timestamp in 1us
- `std::uint16_t exposure_time`
Image exposure time, virtual value in [1, 480]

9.9.1 详细描述

Image data.

9.10 mynteye::device::ImgParams 结构体参考

9.11 mynteye::ImuData 结构体参考

IMU data.

Public 属性

- `std::uint32_t` [frame_id](#)
IMU frame id
- `std::uint8_t` [flag](#)
IMU accel or gyro flag
- `std::uint64_t` [timestamp](#)
IMU timestamp in 1us
- `double` [accel](#) [3]
IMU accelerometer data for 3-axis: X, Y, Z.
- `double` [gyro](#) [3]
IMU gyroscope data for 3-axis: X, Y, Z.
- `double` [temperature](#)
IMU temperature

9.11.1 详细描述

IMU data.

9.11.2 类成员变量说明

9.11.2.1 accel

```
double mynteye::ImuData::accel[3]
```

IMU accelerometer data for 3-axis: X, Y, Z.

9.11.2.2 flag

```
std::uint8_t mynteye::ImuData::flag
```

IMU accel or gyro flag

0: accel and gyro are both valid 1: accel is valid 2: gyro is valid

9.11.2.3 gyro

```
double mynteye::ImuData::gyro[3]
```

IMU gyroscope data for 3-axis: X, Y, Z.

9.12 mynteye::ImuIntrinsics 结构体参考

IMU intrinsics: scale, drift and variances.

Public 属性

- double [scale](#) [3][3]
Scale matrix.
- double [noise](#) [3]
Noise density variances
- double [bias](#) [3]
Random walk variances

9.12.1 详细描述

IMU intrinsics: scale, drift and variances.

9.12.2 类成员变量说明

9.12.2.1 scale

```
double mynteye::ImuIntrinsics::scale[3][3]
```

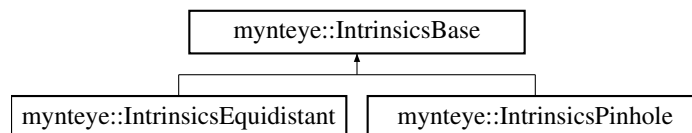
Scale matrix.

```
Scale X   cross axis  cross axis
cross axis Scale Y   cross axis
cross axis cross axis Scale Z
```

9.13 mynteye::device::ImuParams 结构体参考

9.14 mynteye::IntrinsicsBase 结构体参考

类 mynteye::IntrinsicsBase 继承关系图:



Public 成员函数

- [CalibrationModel calib_model \(\) const](#)
The calibration model

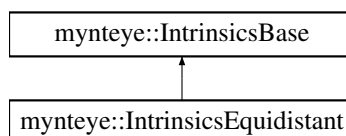
Public 属性

- `std::uint16_t` [width](#)
The width of the image in pixels
- `std::uint16_t` [height](#)
The height of the image in pixels

9.15 mynteye::IntrinsicsEquidistant 结构体参考

Stream intrinsics (Equidistant: KANNALA_BRANDT)

类 mynteye::IntrinsicsEquidistant 继承关系图:



Public 属性

- double [coeffs](#) [8]
The distortion coefficients: k2,k3,k4,k5,mu,mv,u0,v0

额外继承的成员函数

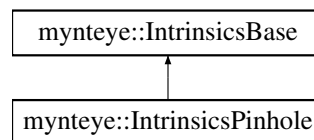
9.15.1 详细描述

Stream intrinsics (Equidistant: KANNALA_BRANDT)

9.16 mynteye::IntrinsicsPinhole 结构体参考

Stream intrinsics (Pinhole)

类 mynteye::IntrinsicsPinhole 继承关系图:



Public 属性

- double [fx](#)
The focal length of the image plane, as a multiple of pixel width
- double [fy](#)
The focal length of the image plane, as a multiple of pixel height
- double [cx](#)
The horizontal coordinate of the principal point of the image
- double [cy](#)
The vertical coordinate of the principal point of the image
- std::uint8_t [model](#)
- double [coeffs](#) [5]
The distortion coefficients: k1,k2,p1,p2,k3

额外继承的成员函数

9.16.1 详细描述

Stream intrinsics (Pinhole)

9.16.2 类成员变量说明

9.16.2.1 model

`std::uint8_t mynteye::IntrinsicsPinhole::model`

弃用 Replaced by `calib_model_`.

The distortion model of the image

9.17 mynteye::device::MotionData 结构体参考

Device motion data.

Public 属性

- `std::shared_ptr< ImuData > imu`
`ImuData.`

9.17.1 详细描述

Device motion data.

9.17.2 类成员变量说明

9.17.2.1 imu

`std::shared_ptr<ImuData> mynteye::device::MotionData::imu`

`ImuData.`

9.18 mynteye::api::MotionData 结构体参考

API motion data.

Public 属性

- `std::shared_ptr< ImuData > imu`
`ImuData`.

9.18.1 详细描述

API motion data.

9.18.2 类成员变量说明

9.18.2.1 imu

`std::shared_ptr<ImuData> mynteye::api::MotionData::imu`

`ImuData`.

9.19 mynteye::MotionIntrinsics 结构体参考

Motion intrinsics, including accelerometer and gyroscope.

Public 属性

- `ImuIntrinsics accel`
Accelerometer intrinsics
- `ImuIntrinsics gyro`
Gyroscope intrinsics

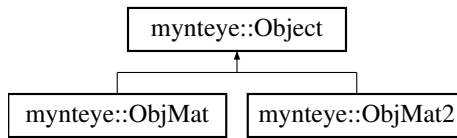
9.19.1 详细描述

Motion intrinsics, including accelerometer and gyroscope.

9.20 mynteye::Object 结构体参考

Input & output object.

类 mynteye::Object 继承关系图:



静态 Public 成员函数

- `template<typename T >`
`static T * Cast (Object *obj)`
 Cast the obj to T pointer
- `template<typename T >`
`static const T * Cast (const Object *obj)`
 Cast the obj to const T pointer

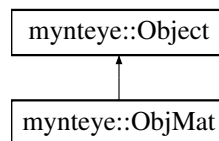
9.20.1 详细描述

Input & output object.

9.21 mynteye::ObjMat 结构体参考

Input & output object of one cv::Mat.

类 mynteye::ObjMat 继承关系图:



Public 属性

- `cv::Mat value`
 The value
- `std::uint16_t id`
 The id
- `std::shared_ptr< ImgData > data`
 The data

额外继承的成员函数

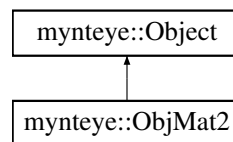
9.21.1 详细描述

Input & output object of one cv::Mat.

9.22 mynteye::ObjMat2 结构体参考

Input & output object of two cv::Mat.

类 mynteye::ObjMat2 继承关系图:



Public 属性

- cv::Mat [first](#)
The first value
- std::uint16_t [first_id](#)
The first id
- std::shared_ptr< [ImgData](#) > [first_data](#)
The first data
- cv::Mat [second](#)
The second value
- std::uint16_t [second_id](#)
The second id
- std::shared_ptr< [ImgData](#) > [second_data](#)
The second data

额外继承的成员函数

9.22.1 详细描述

Input & output object of two cv::Mat.

9.23 mynteye::OptionInfo 结构体参考

Option info.

Public 属性

- `std::int32_t min`
Minimum value
- `std::int32_t max`
Maximum value
- `std::int32_t def`
Default value

9.23.1 详细描述

Option info.

9.24 mynteye::Plugin 类参考

The plugin which could implement processing by yourself.

Public 成员函数

- virtual void `OnCreate` (`API *api`)
Called when plugin created.
- virtual bool `OnRectifyProcess` (`Object *const in, Object *const out`)
Called when process rectify.
- virtual bool `OnDisparityProcess` (`Object *const in, Object *const out`)
Called when process disparity.
- virtual bool `OnDisparityNormalizedProcess` (`Object *const in, Object *const out`)
Called when process normalized disparity.
- virtual bool `OnPointsProcess` (`Object *const in, Object *const out`)
Called when process points.
- virtual bool `OnDepthProcess` (`Object *const in, Object *const out`)
Called when process depth.

9.24.1 详细描述

The plugin which could implement processing by yourself.

9.24.2 成员函数说明

9.24.2.1 OnCreate()

```
virtual void mynteye::Plugin::OnCreate (  
    API * api ) [inline], [virtual]
```

Called when plugin created.

参数

api	the API instacne.
-----	-------------------

9.24.2.2 OnDepthProcess()

```
virtual bool mynteye::Plugin::OnDepthProcess (  
    Object *const in,  
    Object *const out ) [inline], [virtual]
```

Called when process depth.

参数

in	input object.
out	output object.

返回

true if you process depth.

9.24.2.3 OnDisparityNormalizedProcess()

```
virtual bool mynteye::Plugin::OnDisparityNormalizedProcess (  
    Object *const in,  
    Object *const out ) [inline], [virtual]
```

Called when process normalized disparity.

参数

in	input object.
out	output object.

返回

true if you process normalized disparity.

9.24.2.4 OnDisparityProcess()

```
virtual bool mynteye::Plugin::OnDisparityProcess (  
    Object *const in,  
    Object *const out ) [inline], [virtual]
```

Called when process disparity.

参数

in	input object.
out	output object.

返回

true if you process disparity.

9.24.2.5 OnPointsProcess()

```
virtual bool mynteye::Plugin::OnPointsProcess (  
    Object *const in,  
    Object *const out ) [inline], [virtual]
```

Called when process points.

参数

in	input object.
out	output object.

返回

true if you process points.

9.24.2.6 OnRectifyProcess()

```
virtual bool mynteye::Plugin::OnRectifyProcess (  
    Object *const in,  
    Object *const out ) [inline], [virtual]
```

Called when process rectify.

参数

in	input object.
out	output object.

返回

true if you process rectify.

9.25 mynteye::Resolution 结构体参考

[Resolution](#).

Public 属性

- [std::uint16_t width](#)
Width
- [std::uint16_t height](#)
Height

9.25.1 详细描述

[Resolution.](#)

9.26 mynteye::api::StreamData 结构体参考

[API stream data.](#)

Public 属性

- `std::shared_ptr< ImgData > img`
[ImgData.](#)
- `cv::Mat frame`
Frame.
- `std::shared_ptr< device::Frame > frame_raw`
Raw frame.
- `std::uint16_t frame_id`
Frame ID.

9.26.1 详细描述

[API stream data.](#)

9.26.2 类成员变量说明

9.26.2.1 frame

`cv::Mat mynteye::api::StreamData::frame`

Frame.

9.26.2.2 frame_id

`std::uint16_t mynteye::api::StreamData::frame_id`

Frame ID.

9.26.2.3 frame_raw

`std::shared_ptr<device::Frame> mynteye::api::StreamData::frame_raw`

Raw frame.

9.26.2.4 img

`std::shared_ptr<ImgData> mynteye::api::StreamData::img`

[ImgData](#).

9.27 mynteye::device::StreamData 结构体参考

[Device](#) stream data.

Public 属性

- `std::shared_ptr< ImgData > img`
[ImgData](#).
- `std::shared_ptr< Frame > frame`
[Frame](#).
- `std::uint16_t frame_id`
[Frame](#) ID.

9.27.1 详细描述

[Device](#) stream data.

9.27.2 类成员变量说明

9.27.2.1 frame

`std::shared_ptr<Frame> mynteye::device::StreamData::frame`

[Frame](#).

9.27.2.2 frame_id

`std::uint16_t mynteye::device::StreamData::frame_id`

[Frame ID](#).

9.27.2.3 img

`std::shared_ptr<ImgData> mynteye::device::StreamData::img`

[ImgData](#).

9.28 mynteye::StreamRequest 结构体参考

Stream request.

Public 属性

- `std::uint16_t width`
Stream width in pixels
- `std::uint16_t height`
Stream height in pixels
- `Format format`
Stream pixel format
- `std::uint16_t fps`
Stream frames per second

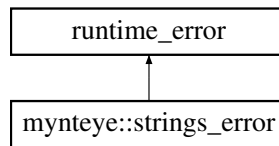
9.28.1 详细描述

Stream request.

9.29 mynteye::strings_error 类参考

The strings error

类 mynteye::strings_error 继承关系图:



9.29.1 详细描述

The strings error

9.30 mynteye::Type 类参考

[Type.](#)

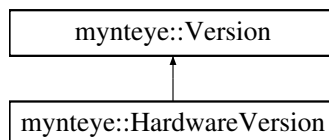
9.30.1 详细描述

[Type.](#)

9.31 mynteye::Version 类参考

[Version.](#)

类 mynteye::Version 继承关系图:



9.31.1 详细描述

[Version.](#)

索引

- accel
 - mynteye::ImuData, [48](#)
- ACCELEROMETER_LOW_PASS_FILTER
 - Enumerations, [27](#)
- ACCELEROMETER_RANGE
 - Enumerations, [27](#)
- AddOns
 - Enumerations, [24](#)
- ALL
 - Enumerations, [28](#)
- BGR888
 - Enumerations, [25](#)
- BRIGHTNESS
 - Enumerations, [26](#)
- CalibrationModel
 - Intrinsics & Extrinsics, [29](#)
- Capabilities
 - Enumerations, [24](#)
- clone
 - mynteye::device::Frame, [46](#)
- COLOR
 - Enumerations, [25](#)
- CONTRAST
 - Enumerations, [26](#)
- Create
 - mynteye::API, [36](#), [37](#)
 - mynteye::Device, [43](#)
- data
 - mynteye::device::Frame, [46](#)
- Datatypes, [31](#)
- DEPTH
 - Enumerations, [25](#), [28](#)
- DESIRED_BRIGHTNESS
 - Enumerations, [27](#)
- DEVICE_NAME
 - Enumerations, [26](#)
- devices
 - mynteye::Context, [39](#)
- DISPARITY
 - Enumerations, [28](#)
- DISPARITY_NORMALIZED
 - Enumerations, [28](#)
- EnableStreamData
 - mynteye::API, [38](#)
- Enumerations, [23](#)
 - ACCELEROMETER_LOW_PASS_FILTER, [27](#)
 - ACCELEROMETER_RANGE, [27](#)
 - AddOns, [24](#)
 - ALL, [28](#)
 - BGR888, [25](#)
 - BRIGHTNESS, [26](#)
 - Capabilities, [24](#)
 - COLOR, [25](#)
 - CONTRAST, [26](#)
 - DEPTH, [25](#), [28](#)
 - DESIRED_BRIGHTNESS, [27](#)
 - DEVICE_NAME, [26](#)
 - DISPARITY, [28](#)
 - DISPARITY_NORMALIZED, [28](#)
 - ERASE_CHIP, [27](#)
 - EXPOSURE_MODE, [27](#)
 - FIRMWARE_VERSION, [26](#)
 - FISHEYE, [25](#)
 - Format, [25](#)
 - FRAME_RATE, [27](#)
 - GAIN, [26](#)
 - GREY, [25](#)
 - GYROSCOPE_LOW_PASS_FILTER, [27](#)

- GYROSCOPE_RANGE, 27
- HARDWARE_VERSION, 26
- HDR_MODE, 27
- IMU, 25
- IMU_FREQUENCY, 27
- IMU_TYPE, 26
- Info, 25
- INFRARED, 24, 25
- INFRARED2, 24, 25
- IR_CONTROL, 27
- LAST, 24–28
- LEFT, 28
- LEFT_RECTIFIED, 28
- LENS_TYPE, 26
- MAX_EXPOSURE_TIME, 27
- MAX_GAIN, 27
- MIN_EXPOSURE_TIME, 27
- Model, 26
- MOTION_TRACKING, 28
- NOMINAL_BASELINE, 26
- Option, 26
- POINTS, 25, 28
- RGB888, 25
- RIGHT, 28
- RIGHT_RECTIFIED, 28
- SERIAL_NUMBER, 26
- Source, 27
- SPEC_VERSION, 26
- STANDARD, 26
- STANDARD2, 26
- STANDARD210A, 26
- STEREO, 25
- STEREO_COLOR, 25
- Stream, 28
- VIDEO_STREAMING, 28
- YUYV, 25
- ZERO_DRIFT_CALIBRATION, 27
- ERASE_CHIP
 - Enumerations, 27
- EXPOSURE_MODE
 - Enumerations, 27
- FIRMWARE_VERSION
 - Enumerations, 26
- FISHEYE
 - Enumerations, 25
- flag
 - mynteye::ImuData, 48
- Format
 - Enumerations, 25
- format
 - mynteye::device::Frame, 46
- frame
 - mynteye::api::StreamData, 61
 - mynteye::device::StreamData, 63
- frame_id
 - mynteye::api::StreamData, 61
 - mynteye::device::StreamData, 63
- FRAME_RATE
 - Enumerations, 27
- frame_raw
 - mynteye::api::StreamData, 62
- GAIN
 - Enumerations, 26
- get_real_exposure_time
 - Utilities, 21
- GetIntrinsics
 - mynteye::API, 38
- GetLatestStreamData
 - mynteye::Device, 43
- GetStreamDatas
 - mynteye::API, 38
 - mynteye::Device, 43
- GREY
 - Enumerations, 25
- gyro
 - mynteye::ImuData, 49
- GYROSCOPE_LOW_PASS_FILTER
 - Enumerations, 27
- GYROSCOPE_RANGE
 - Enumerations, 27
- HARDWARE_VERSION
 - Enumerations, 26
- HDR_MODE
 - Enumerations, 27

- height
 - mynteye::device::Frame, [46](#)
- img
 - mynteye::api::StreamData, [62](#)
 - mynteye::device::StreamData, [63](#)
- IMU
 - Enumerations, [25](#)
- imu
 - mynteye::api::MotionData, [53](#)
 - mynteye::device::MotionData, [52](#)
- IMU_FREQUENCY
 - Enumerations, [27](#)
- IMU_TYPE
 - Enumerations, [26](#)
- Info
 - Enumerations, [25](#)
- INFRARED
 - Enumerations, [24](#), [25](#)
- INFRARED2
 - Enumerations, [24](#), [25](#)
- Intrinsics & Extrinsics, [29](#)
 - CalibrationModel, [29](#)
 - KANNALA_BRANDT, [30](#)
 - PINHOLE, [30](#)
 - UNKNOWN, [30](#)
- Inverse
 - mynteye::Extrinsics, [44](#)
- IR_CONTROL
 - Enumerations, [27](#)
- KANNALA_BRANDT
 - Intrinsics & Extrinsics, [30](#)
- LAST
 - Enumerations, [24-28](#)
- LEFT
 - Enumerations, [28](#)
- LEFT_RECTIFIED
 - Enumerations, [28](#)
- LENS_TYPE
 - Enumerations, [26](#)
- MAX_EXPOSURE_TIME
 - Enumerations, [27](#)
- MAX_GAIN
 - Enumerations, [27](#)
- MIN_EXPOSURE_TIME
 - Enumerations, [27](#)
- Model
 - Enumerations, [26](#)
- model
 - mynteye::IntrinsicsPinhole, [52](#)
- motion_callback_t
 - mynteye::API, [36](#)
 - mynteye::Device, [42](#)
- MOTION_TRACKING
 - Enumerations, [28](#)
- mynteye::API, [33](#)
 - Create, [36](#), [37](#)
 - EnableStreamData, [38](#)
 - GetIntrinsics, [38](#)
 - GetStreamDatas, [38](#)
 - motion_callback_t, [36](#)
 - stream_callback_t, [36](#)
- mynteye::api::MotionData, [53](#)
 - imu, [53](#)
- mynteye::api::StreamData, [61](#)
 - frame, [61](#)
 - frame_id, [61](#)
 - frame_raw, [62](#)
 - img, [62](#)
- mynteye::AsyncCallback< Data >, [39](#)
- mynteye::Context, [39](#)
 - devices, [39](#)
- mynteye::Device, [39](#)
 - Create, [43](#)
 - GetLatestStreamData, [43](#)
 - GetStreamDatas, [43](#)
 - motion_callback_t, [42](#)
 - stream_callback_t, [42](#)
- mynteye::device::Frame, [45](#)
 - clone, [46](#)
 - data, [46](#)
 - format, [46](#)
 - height, [46](#)
 - size, [46](#)

- width, [47](#)
- mynteye::device::ImgParams, [48](#)
- mynteye::device::ImuParams, [50](#)
- mynteye::device::MotionData, [52](#)
 - imu, [52](#)
- mynteye::device::StreamData, [62](#)
 - frame, [63](#)
 - frame_id, [63](#)
 - img, [63](#)
- mynteye::DeviceInfo, [44](#)
- mynteye::Extrinsics, [44](#)
 - Inverse, [44](#)
- mynteye::HardwareVersion, [47](#)
- mynteye::ImgData, [47](#)
- mynteye::ImuData, [48](#)
 - accel, [48](#)
 - flag, [48](#)
 - gyro, [49](#)
- mynteye::ImuIntrinsics, [49](#)
 - scale, [49](#)
- mynteye::IntrinsicsBase, [50](#)
- mynteye::IntrinsicsEquidistant, [50](#)
- mynteye::IntrinsicsPinhole, [51](#)
 - model, [52](#)
- mynteye::MotionIntrinsics, [53](#)
- mynteye::Object, [54](#)
- mynteye::ObjMat, [54](#)
- mynteye::ObjMat2, [55](#)
- mynteye::OptionInfo, [56](#)
- mynteye::Plugin, [56](#)
 - OnCreate, [57](#)
 - OnDepthProcess, [57](#)
 - OnDisparityNormalizedProcess, [57](#)
 - OnDisparityProcess, [58](#)
 - OnPointsProcess, [58](#)
 - OnRectifyProcess, [60](#)
- mynteye::Resolution, [60](#)
- mynteye::StreamRequest, [63](#)
- mynteye::strings_error, [64](#)
- mynteye::Type, [64](#)
- mynteye::Version, [64](#)
- NOMINAL_BASELINE
 - Enumerations, [26](#)
- OnCreate
 - mynteye::Plugin, [57](#)
- OnDepthProcess
 - mynteye::Plugin, [57](#)
- OnDisparityNormalizedProcess
 - mynteye::Plugin, [57](#)
- OnDisparityProcess
 - mynteye::Plugin, [58](#)
- OnPointsProcess
 - mynteye::Plugin, [58](#)
- OnRectifyProcess
 - mynteye::Plugin, [60](#)
- Option
 - Enumerations, [26](#)
- PINHOLE
 - Intrinsics & Extrinsics, [30](#)
- POINTS
 - Enumerations, [25](#), [28](#)
- RGB888
 - Enumerations, [25](#)
- RIGHT
 - Enumerations, [28](#)
- RIGHT_RECTIFIED
 - Enumerations, [28](#)
- scale
 - mynteye::ImuIntrinsics, [49](#)
- select
 - Utilities, [22](#)
- select_request
 - Utilities, [22](#)
- SERIAL_NUMBER
 - Enumerations, [26](#)
- size
 - mynteye::device::Frame, [46](#)
- Source
 - Enumerations, [27](#)
- SPEC_VERSION
 - Enumerations, [26](#)
- STANDARD

- Enumerations, [26](#)
- STANDARD2
 - Enumerations, [26](#)
- STANDARD210A
 - Enumerations, [26](#)
- STEREO
 - Enumerations, [25](#)
- STEREO_COLOR
 - Enumerations, [25](#)
- Stream
 - Enumerations, [28](#)
- stream_callback_t
 - mynteye::API, [36](#)
 - mynteye::Device, [42](#)
- UNKNOWN
 - Intrinsics & Extrinsics, [30](#)
- Utilities, [21](#)
 - get_real_exposure_time, [21](#)
 - select, [22](#)
 - select_request, [22](#)
- VIDEO_STREAMING
 - Enumerations, [28](#)
- width
 - mynteye::device::Frame, [47](#)
- YUYV
 - Enumerations, [25](#)
- ZERO_DRIFT_CALIBRATION
 - Enumerations, [27](#)