

MYNT EYE SDK

v2.0

制作者 Doxygen 1.8.14

Contents

1	MYNT EYE SDK	1
2	SDK 指导	3
2.1	编译 on Linux	3
2.2	样例	4
2.3	工具	5
2.4	日志	6
2.5	OpenCV 依赖	6
2.6	ROS 封装	6
3	设备数据说明	7
3.1	硬件信息说明	7
3.2	图像参数说明	7
3.3	IMU 参数说明	8
3.4	图像数据说明	9
3.5	IMU 数据说明	9
4	设备控制说明	11
4.1	控制 API 说明	11
4.2	拓展通道说明	13
5	模块索引	15
5.1	模块	15
6	类索引	17
6.1	类列表	17

7	模块说明	19
7.1	Enumerations	19
7.1.1	详细描述	20
7.1.2	枚举类型说明	20
7.1.2.1	Capabilities	20
7.1.2.2	Format	20
7.1.2.3	Info	20
7.1.2.4	Model	21
7.1.2.5	Option	21
7.1.2.6	Source	22
7.1.2.7	Stream	22
7.2	Intrinsics & Extrinsics	23
7.2.1	详细描述	23
7.3	Datatypes	24
7.3.1	详细描述	24
8	类说明	25
8.1	Extrinsics结构体 参考	25
8.1.1	详细描述	25
8.2	glog_init结构体 参考	25
8.3	ImgData结构体 参考	25
8.3.1	详细描述	26
8.4	ImgExtrinsics结构体 参考	26
8.4.1	详细描述	26
8.5	ImgIntrinsics结构体 参考	26
8.5.1	详细描述	26
8.6	ImuData结构体 参考	26
8.6.1	详细描述	27
8.6.2	类成员变量说明	27
8.6.2.1	accel	27
8.6.2.2	gyro	27
8.7	ImuExtrinsics结构体 参考	27
8.7.1	详细描述	27
8.8	ImuIntrinsics结构体 参考	28
8.8.1	详细描述	28
8.9	ImuSensorIntrinsics结构体 参考	28
8.9.1	详细描述	28
8.10	OptionInfo结构体 参考	28
8.10.1	详细描述	28
8.11	StreamRequest结构体 参考	28
8.11.1	详细描述	29
	索引	31

Chapter 1

MYNT EYE SDK

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

Chapter 2

SDK 指导

- [编译 on Linux](#)
- [样例](#)
- [工具](#)
- [日志](#)
- [OpenCV 依赖](#)
- [ROS 封装](#)

2.1 编译 on Linux

获取代码

```
git clone https://github.com/slightech/MYNT-EYE-SDK-2.git
```

准备依赖

```
cd mynt-eye-sdk-2/  
make init
```

编译代码

```
make install
```

结果:

```
└─ mynt-eye-sdk-2
  └─ _build
  └─ _install
    └─ include
      └─ mynteye
    └─ lib
      └─ cmake
        ├── mynteye-config-version.cmake
        ├── mynteye-config.cmake
        ├── mynteye-targets-release.cmake
        ├── mynteye-targets.cmake
        ├── libmynteye.2.0.0.dylib
        ├── libmynteye.2.dylib
        └── libmynteye.dylib
      └─ _output
```

CMake 如何引入编译好的库，可参考 `samples/CMakeLists.txt` 里的配置。

2.2 样例

样例在 `<sdk>/samples` 目录，其提供了不同接口层的使用范例。

- `api`，API 层接口，依赖 `OpenCV`。
- `device`，中间层接口，不依赖 `OpenCV`。
- `uvc`，底层接口。

目前 API 层接口，正在实现中。

依赖

- `OpenCV`，用于显示图像。

编译

```
make samples
```


运行

运行 device 层接口样例，显示图像并输出 IMU。

```
./samples/_output/bin/device/camera_d
```

2.3 工具

样例在 <sdk>/tools 目录，其提供一些有用的工具。

准备

```
make tools
cd tools/
sudo pip install -r requirements.txt
```

录制数据集

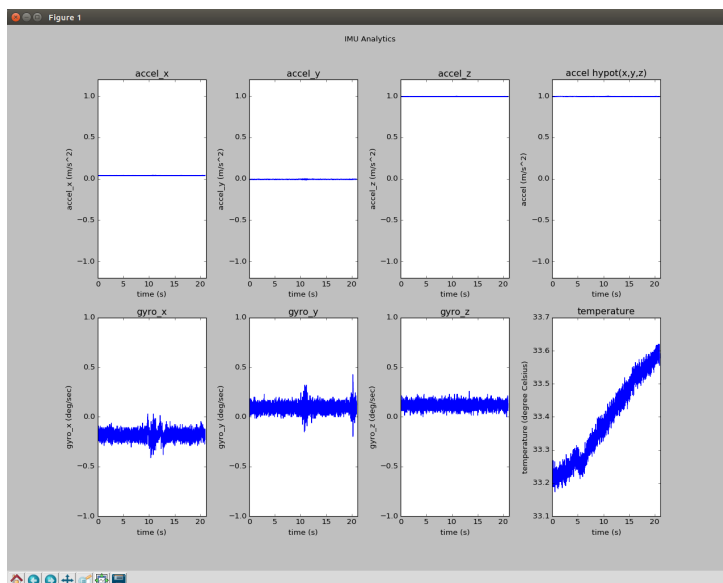
```
./tools/_output/bin/dataset/record
```

默认录制进 dataset 目录，加参数可指定该目录。

分析数据集

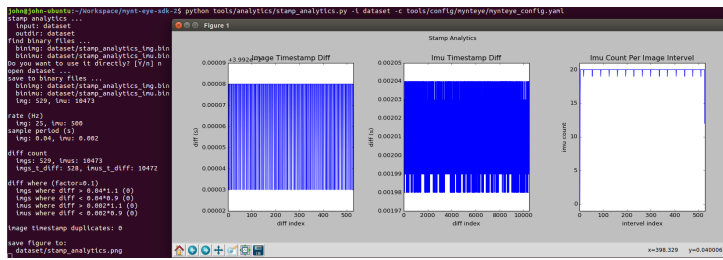
分析 IMU 数据，

```
python tools/analytics/imu_analytics.py -i dataset -c tools/config/mynteye/mynteye_config.yaml \
-al=-1.2,1.2 -gl= -gdu=d -gsu=d -kl=
```



分析图像 & IMU 时间戳，

```
python tools/analytics/stamp_analytics.py -i dataset -c tools/config/mynteye/mynteye_config.yaml
```



结语

如果用 ROS，分析脚本也支持 ROS Bag 格式。关于 ROS 的录制和分析，请见 `tools/README.md`。

2.4 日志

日志系统用的 `glog`，通用配置在头文件 `glog_init.h` 里。

- 日志文件会存储在当前工作目录，`make cleanlog` 可以清理。
- 如果需要打开详细日志，请取消 `glog_init.h` 里注释的 `FLAGS_v = 2;`，重新编译。

2.5 OpenCV 依赖

目前 SDK 本身未依赖 OpenCV，仅是样例依赖其显示图像。

2.6 ROS 封装

ROS 封装实现在 `<sdk>/wrappers/ros` 目录。

依赖

- ROS 环境。

编译

```
cd <sdk>
make ros
```

运行

运行发布节点，

```
source wrappers/ros/devel/setup.bash
roslaunch mynt_eye_ros_wrapper mynteye.launch
```

运行发布节点，同时打开 RViz 预览图像，

```
source wrappers/ros/devel/setup.bash
roslaunch mynt_eye_ros_wrapper display.launch
```

Chapter 3

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](#)

3.1 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-EYE-?	✓	✓	16	MYNT-EYE-↔ S1000
序列号	serial_number	-	✓	✓	16	
固件版本	firmware_↔ version	-	✓	✓	2	major,minor
硬件版本	hardware_↔ version	-	×	✓	3	major,minor,flag
协议版本	spec_version	-	×	✓	2	major,minor
镜头类型	lens_type	-	×	✓	4	vendor(2),product(2)
IMU 类型	imu_type	-	×	✓	4	vendor(2),product(2)
基线长度	nominal_↔ baseline	-	×	✓	2	单位 mm

- 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。

3.2 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

图像内参不同分辨率会不同。如果多分辨率的话，就会有多个。

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

3.3 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

3.4 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	10 us	4	uint32_t
曝光时间	exposure_time	10 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	ExposureTime	Checksum
字节数	1	1	2	4	2	1
类型	uint8↔ _t	uint8_t	uint16_t	uint32_t	uint16_t	uint8_t
描述	0x3B	0x08 (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：0.01 ms / 10 us 。
 - 4 字节能表示的最大时间约是 11.9 小时，溢出后将重累计。
- 时间累计是从上电时从开始，而不是从打开时开始。

3.5 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多个 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	Serial Number	Timestamp	Count	IMU Datas
字节数	4	4	1	...
类型	uint32_t	uint32_t	uint8_t	-
描述	序列号	IMU 基准时间戳	IMU 段数量	所包含的 IMU 段

IMU 段

Name	Offset	FrameID	Accelerometer	Temperature	Gyroscope
字节数	2	2	6	2	6
类型	int16_t	uint16_t	int16_t * 3	int16_t	int16_t * 3
Description	相对基准时间戳的偏移量	图像帧 ID	加速度计 x y z 三轴的值	IMU 的温度	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 **8 g**，陀螺仪量程默认值为 **1000 deg/s²**。
- 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 4

设备控制说明

- [控制 API 说明](#)
- [拓展通道说明](#)

4.1 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
增益	gain	2	24	0	48	✓	0x12	关闭自动曝光，手动设定的参数
亮度	brightness/exposure_time	2	120	0	240	✓	0x14	关闭自动曝光，手动设定的参数
对比度	contrast/black_level_calibration	2	127	0	255	✓	0x10	

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
图像帧率	frame↔ _rate	2	25	10	60	✓	0x21	XU↔ CA↔ M_C↔ TRL	0x0100	步 进 为5, 即 有 效 值 为{10,15,20,25,30,35,40,45,50,55,60}
IMU 频率	imu↔ frequency	2	200	100	500	✓	0x23	XU↔ CA↔ M_C↔ TRL	0x0100	有 效 值 为{100,200,250,333,500}
曝光模式	exposure↔ _mode	1	0	0	1	✓	0x0F	XU↔ CA↔ M_C↔ TRL	0x0100	0: 开 自 曝 光 ; 关 闭 1: 关 闭
最大增益	max↔ _gain	2	48	0	48	✓	0x1D	XU↔ CA↔ M_C↔ TRL	0x0100	开 始 自 曝 光 , 可 定 义 的 阈 值
最大曝光时间	max↔ _↔ exposure↔ _time	2	240	0	240	✓	0x1B	XU↔ CA↔ M_C↔ TRL	0x0100	开 始 自 曝 光 , 可 定 义 的 阈 值
期望亮度	desired↔ _↔ brightness	2	192	0	255	✓	0x19	XU↔ CA↔ M_C↔ TRL	0x0100	
IR 控制	ir↔ control	1	0	0	160	×	-	XU↔ CA↔ M_C↔ TRL	0x0100	
HDR 模式	hdr↔ mode	1	0	0	1	✓	0x1F	XU↔ CA↔ M_C↔ TRL	0x0100	0: 10-bit; 1: 12-bit
零漂标定	zero↔ _↔ drift↔ _↔ calibration		-	-	-	×	-	XU↔ HAL↔ F_D↔ UPL↔ EX	0x0200	
擦除芯片	erase↔ _chip		-	-	-	×	-	XU↔ HAL↔ F_D↔ UPL↔ EX	0x0200	

4.2 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	0x0100	3	
半双工通道	XU_HALF_DUPLEX_CHANNEL	0x0200	20	
IMU 请求通道	XU_IMUDATA_WRITE_CHANNEL	0x0300	5	
IMU 响应通道	XU_IMUDATA_READ_CHANNEL	0x0400	2000	
文件通道	XU_FILE_CHANNEL	0x0500	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 Get 或 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	Checks
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	Description
0	0: get; 1: set
1~4	未定义
5	IMU 参数
6	图像参数
7	硬件信息

文件内容包

Name	ID	Size	Content
字节数	1	2	-

Name	ID	Size	Content
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	0	256
图像参数	1	256
IMU 参数	2	512

Chapter 5

模块索引

5.1 模块

这里列出了所有模块:

Enumerations	19
Intrinsics & Extrinsics	23
Datatypes	24

Chapter 6

类索引

6.1 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：

Extrinsics	
Extrinsics , represent how the different datas are connected	25
glog_init	25
ImgData	
Image data	25
ImgExtrinsics	
Image extrinsics	26
ImgIntrinsics	
Image intrinsics	26
ImuData	
IMU data	26
ImuExtrinsics	
IMU extrinsics	27
ImuIntrinsics	
IMU intrinsics, including accelerometer and gyroscope	28
ImuSensorIntrinsics	
IMU sensor intrinsics: scale, drift and variances	28
OptionInfo	
Option info	28
StreamRequest	
Stream request	28

Chapter 7

模块说明

7.1 Enumerations

Public enumeration types.

枚举

- enum `Model` : `std::uint8_t` { `Model::STANDARD`, `Model::LAST` }
Device model.
- enum `Stream` : `std::uint8_t` {
`Stream::LEFT`, `Stream::RIGHT`, `Stream::LEFT_RECTIFIED`, `Stream::RIGHT_RECTIFIED`,
`Stream::DISPARITY`, `Stream::DISPARITY_NORMALIZED`, `Stream::DEPTH`, `Stream::POINTS`,
`Stream::LAST` }
Streams define different type of data.
- enum `Capabilities` : `std::uint8_t` {
`Capabilities::STEREO`, `Capabilities::COLOR`, `Capabilities::DEPTH`, `Capabilities::POINTS`,
`Capabilities::FISHEYE`, `Capabilities::INFRARED`, `Capabilities::INFRARED2`, `Capabilities::IMU`,
`Capabilities::LAST` }
Capabilities define the full set of functionality that the device might provide.
- enum `Info` : `std::uint8_t` {
`Info::DEVICE_NAME`, `Info::SERIAL_NUMBER`, `Info::FIRMWARE_VERSION`, `Info::HARDWARE_VERSION`,
`Info::SPEC_VERSION`, `Info::LENS_TYPE`, `Info::IMU_TYPE`, `Info::NOMINAL_BASELINE`,
`Info::LAST` }
Camera info fields are read-only strings that can be queried from the device.
- enum `Option` : `std::uint8_t` {
`Option::GAIN`, `Option::BRIGHTNESS`, `Option::CONTRAST`, `Option::FRAME_RATE`,
`Option::IMU_FREQUENCY`, `Option::EXPOSURE_MODE`, `Option::MAX_GAIN`, `Option::MAX_EXPOSURE_TIME`,
`Option::DESIRED_BRIGHTNESS`, `Option::IR_CONTROL`, `Option::HDR_MODE`, `Option::ZERO_DRIFT_CALIBRATION`,
`Option::ERASE_CHIP`, `Option::LAST` }
Camera control options define general configuration controls.
- enum `Source` : `std::uint8_t` { `Source::VIDEO_STREAMING`, `Source::MOTION_TRACKING`, `Source::ALL`,
`Source::LAST` }
Source allows the user to choose which data to be captured.
- enum `Format` : `std::uint32_t` { `Format::GREY` = $((\text{std::uint32_t}('G') \ll 8) \mid (\text{std::uint32_t}('R') \ll 8) \mid (\text{std::uint32_t}('B') \ll 8) \mid (\text{std::uint32_t}('A') \ll 8))$, `Format::YUYV` = $((\text{std::uint32_t}('Y') \ll 24) \mid (\text{std::uint32_t}('U') \ll 8) \mid (\text{std::uint32_t}('Y') \ll 16) \mid (\text{std::uint32_t}('V') \ll 24))$, `Format::LAST` }
Formats define how each stream can be encoded.

7.1.1 详细描述

Public enumeration types.

7.1.2 枚举类型说明

7.1.2.1 Capabilities

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

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

枚举值

STEREO	Provides stereo 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

7.1.2.2 Format

```
enum 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
LAST	Last guard

7.1.2.3 Info

```
enum 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

7.1.2.4 Model

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

Device model.

枚举值

STANDARD	Standard
LAST	Last guard

7.1.2.5 Option

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

Camera control options define general configuration controls.

枚举值

GAIN	Image gain, setting it if manual-exposure
BRIGHTNESS	Image brightness, setting it if manual-exposure
CONTRAST	Image contrast
FRAME_RATE	Image frame rate
IMU_FREQUENCY	IMU frequency
EXPOSURE_MODE	Exposure mode 0: enable auto-exposure 1: disable auto-exposure (manual-exposure)
MAX_GAIN	Max gain, setting it if auto-exposure
MAX_EXPOSURE_TIME	Max exposure time, setting it if auto-exposure
DESIRED_BRIGHTNESS	Desired brightness
IR_CONTROL	IR control
HDR_MODE	HDR mode 0: 10-bit 1: 12-bit
ZERO_DRIFT_CALIBRATION	Zero drift calibration
ERASE_CHIP	Erase chip
LAST	Last guard

制作者 Doxygen

7.1.2.6 Source

```
enum 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

7.1.2.7 Stream

```
enum 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

7.2 Intrinsic & Extrinsic

Intrinsic and extrinsic properties.

类

- struct [ImgIntrinsic](#)
Image intrinsic.
- struct [ImuSensorIntrinsic](#)
IMU sensor intrinsic: scale, drift and variances.
- struct [ImuIntrinsic](#)
IMU intrinsic, including accelerometer and gyroscope.
- struct [Extrinsic](#)
Extrinsic, represent how the different datas are connected.
- struct [ImgExtrinsic](#)
Image extrinsic.
- struct [ImuExtrinsic](#)
IMU extrinsic.

7.2.1 详细描述

Intrinsic and extrinsic properties.

7.3 Datatypes

Public data types.

类

- struct [ImgData](#)
Image data.
- struct [ImuData](#)
IMU data.
- struct [OptionInfo](#)
Option info.

7.3.1 详细描述

Public data types.

Chapter 8

类说明

8.1 Extrinsics结构体 参考

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

Public 属性

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

8.1.1 详细描述

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

8.2 glog_init结构体 参考

8.3 ImgData结构体 参考

Image data.

Public 属性

- std::uint16_t [frame_id](#)
Image frame id
- std::uint32_t [timestamp](#)
Image timestamp in 0.01ms
- std::uint16_t [exposure_time](#)
Image exposure time in 0.01ms

8.3.1 详细描述

Image data.

8.4 ImgExtrinsics结构体 参考

Image extrinsics.

8.4.1 详细描述

Image extrinsics.

8.5 ImgIntrinsics结构体 参考

Image intrinsics.

Public 属性

- `std::uint16_t width`
width of the image in pixels
- `std::uint16_t height`
height of the image in pixels
- `double fx`
focal length of the image plane, as a multiple of pixel width
- `double fy`
focal length of the image plane, as a multiple of pixel height
- `double cx`
horizontal coordinate of the principal point of the image
- `double cy`
vertical coordinate of the principal point of the image
- `std::uint8_t model`
distortion model of the image
- `double coeffs [5]`
distortion coefficients: k_1, k_2, p_1, p_2, k_3

8.5.1 详细描述

Image intrinsics.

8.6 ImuData结构体 参考

IMU data.

Public 属性

- `std::uint16_t frame_id`
Image frame id
- `std::uint32_t timestamp`
IMU timestamp in 0.01ms
- `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

8.6.1 详细描述

IMU data.

8.6.2 类成员变量说明

8.6.2.1 accel

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

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

8.6.2.2 gyro

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

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

8.7 ImuExtrinsics结构体 参考

IMU extrinsics.

8.7.1 详细描述

IMU extrinsics.

8.8 ImuIntrinsics结构体 参考

IMU intrinsics, including accelerometer and gyroscope.

8.8.1 详细描述

IMU intrinsics, including accelerometer and gyroscope.

8.9 ImuSensorIntrinsics结构体 参考

IMU sensor intrinsics: scale, drift and variances.

Public 属性

- double [scale](#) [3][3]
Scale X cross axis cross axis cross axis Scale Y cross axis cross axis cross axis Scale Z
- double [noise](#) [3]
Noise density variances
- double [bias](#) [3]
Random walk variances

8.9.1 详细描述

IMU sensor intrinsics: scale, drift and variances.

8.10 OptionInfo结构体 参考

Option info.

8.10.1 详细描述

Option info.

8.11 StreamRequest结构体 参考

Stream request.

Public 属性

- `std::uint16_t width`
width in pixels
- `std::uint16_t height`
height in pixels
- `Format format`
stream pixel format
- `std::uint16_t fps`
frames per second (unused)

8.11.1 详细描述

Stream request.

Index

- accel
 - [ImuData, 27](#)
- Capabilities
 - [Enumerations, 20](#)
- [Datatypes, 24](#)
- Enumerations, [19](#)
 - [Capabilities, 20](#)
 - [Format, 20](#)
 - [Info, 20](#)
 - [Model, 21](#)
 - [Option, 21](#)
 - [Source, 22](#)
 - [Stream, 22](#)
- [Extrinsics, 25](#)
- Format
 - [Enumerations, 20](#)
- [glog_init, 25](#)
- gyro
 - [ImuData, 27](#)
- [ImgData, 25](#)
- [ImgExtrinsics, 26](#)
- [ImgIntrinsics, 26](#)
- [ImuData, 26](#)
 - [accel, 27](#)
 - [gyro, 27](#)
- [ImuExtrinsics, 27](#)
- [ImuIntrinsics, 28](#)
- [ImuSensorIntrinsics, 28](#)
- Info
 - [Enumerations, 20](#)
- [Intrinsics & Extrinsics, 23](#)
- Model
 - [Enumerations, 21](#)
- Option
 - [Enumerations, 21](#)
- [OptionInfo, 28](#)
- Source
 - [Enumerations, 22](#)
- Stream
 - [Enumerations, 22](#)
- [StreamRequest, 28](#)