

SDK

设备上电会开启一个服务器，通过http协议进行参数读写及流数据获取，也可以通过ros主从机通信来控制以及获取数据和设备状态。

1.设置和操作

(1) 设置设备ip

设置后重启生效

(2) tof开关 ON/OFF

通过通信协议中的smart参数设置

默认：OFF

(3) raw_data输出设置

通过通信协议中的smart参数设置

图像：0/1/2/3 (注：0：无图像通过流数据输出；1：左目；2：右目；3：双目图像通过流数据输出)

imu：ON/OFF

tof深度图：ON/OFF

tof幅度图：ON/OFF

默认：OFF

ROS控制需要发送下面的话题，话题数据是自定义的，详情参考配套demo中的msg

```
Type: system_ctrl::viobot_ctrl
Topic: /system_ctrl
```

(4) VIO算法

启停：ON/OFF

重启：restart

重置：reset

重定位: relocation

ROS控制需要发送下面的话题，话题数据是自定义的，详情参考配套demo中的msg

```
Type: system_ctrl::algo_ctrl
Topic: /stereo1_ctrl    /stereo2_ctrl    /mono_ctrl
```

2.数据输出

(1) raw_data

注：需设置raw_data输出

1) imu

```
typedef struct {  
    double acc_x, acc_y, acc_z, groy_x, groy_y, groy_z;  
    uint32_t time_stamp;  
}imu;
```

2) 图像数据

```
uint16_t width,height;  
uint32_t time_stamp;  
char data[width * height];
```

3) tof数据

深度图

```
uint16_t width,height;  
uint32_t time_stamp;  
uint16_t data[width * height];
```

幅度图

```
uint16_t width,height;  
uint32_t time_stamp;  
uint16_t data[width * height];
```

注：默认可以通过ros获取数据

ros话题：

i.图像数据

```
Type: sensor_msgs::Image  
Topic: /image_left    /image_right
```

ii.imu数据

```
Type: sensor_msgs::Imu  
Topic: /imu
```

iii.tof点云数据(需开启tof)

```
Type: sensor_msgs::PointCloud2  
Topic: /tof_cloud  
(x,y,z)
```

iiii.tof幅度图和深度图(需开启tof)

```
Type: sensor_msgs::Image
Topic: /amp_image /depth_image
```

iiii.系统状态

```
#此项为自定义的ros msg,可以在SDK例程里面找到
Type: system_ctrl::viobot_ctrl
Topic: /sys_status
```

iiiii.算法状态

```
#此项为自定义的ros msg,可以在SDK例程里面找到，具体状态见（4）系统算法状态
Type: system_ctrl::algo_staus
Topic: /algo_status
```

iiiiii.相机参数

```
Type: sensor_msgs::CameraInfo
Topic: /camera_left_info /camera_right_info
```

(2) 算法数据

注：启动算法后才会输出

位姿

3行4列变换矩阵 12个float数据 每4个值表示矩阵的一行

说明：该矩阵 $T = [R \ t]$

其中 R (3×3):旋转变换 t (3×1):平移变换

ROS输出：

i.stereo1

pose:

```
Type: nav_msgs::Odometry
Topic: /pr_loop/odometry_rect
```

ii.stereo2

pose:

```
Type: nav_msgs::Odometry
Topic: /pr_loop/odometry_rect
```

pointcloud:

```
Type: sensor_msgs::PointCloud
Topic: /pr_loop/point_cloud_loop_rect
```

iii.mono1

pose:

```
Type: nav_msgs::Odometry
Topic: /mono1_loop/loop_pose
```

(3) 参数数据

按需获取的数据，单次响应

1) 可见光内参

```
//内参结构体
typedef struct{
    float focal_length_x;           // 焦距长度x（像素）
    float focal_length_y;           // 焦距长度y（像素）
    float optical_center_point_x;    // 光心投影坐标x(像素y)
    float optical_center_point_y;    // 光心投影坐标y(像素)
    float radia_distortion_coef_k1;  // radtan相机畸变模型畸变系数k1
    float radia_distortion_coef_k2;  // radtan相机畸变模型畸变系数k2
    float tangential_distortion_p1;  // radtan相机畸变模型畸变系数p1
    float tangential_distortion_p2;  // radtan相机畸变模型畸变系数p2
} LensPara;
```

2) IMU内参

```
//内参结构体
typedef struct{
    float acc_n;                    // 加速度计噪声
    float acc_w;                    // 加速度计随机游走
    float gyr_n;                    // 陀螺仪噪声
    float gyr_w;                    // 陀螺仪随机游走
} ImuParam;
```

3) tof内参

```
//内参结构体
typedef struct{
    float focal_length_x;           // 焦距长度x（像素）
    float focal_length_y;           // 焦距长度y（像素）
    float optical_center_point_x;    // 光心投影坐标x(像素y)
    float optical_center_point_y;    // 光心投影坐标y(像素)
    float radia_distortion_coef_k1;  // radtan相机畸变模型畸变系数k1
    float radia_distortion_coef_k2;  // radtan相机畸变模型畸变系数k2
    float tangential_distortion_p1;  // radtan相机畸变模型畸变系数p1
    float tangential_distortion_p2;  // radtan相机畸变模型畸变系数p2
} LensPara;
```

4) 外参

3行4列变换矩阵 12个float数据

CamToImu

TofToImu

(4) 系统算法状态

```
typedef enum{
    ready = 0,
    stereo1_initializing,
    stereo1_running,
    stereo2_initializing,
    stereo2_running,
    mono_initializing,
    mono_running,
}system_status;
```

单个字节，上电连接后默认为ready，给设备发送了启动指后会进入初始化状态（initializing），初始化完成进入运行状态（running）

3.通信协议

【*协议说明*】 HTTP协议主要用于参数读写及流数据获取，默认端口8000

【*接口说明*】

(1) 网络参数

URL	http://< ip >:< port >/System/network
METHOD	GET/PUT
BODY	{ "ipaddr":"192.168.1.100", "submask":"192.168.0.1.0", "gateway":"192.168.1.1", "macaddr":"FF:FF:FF:FF:FF:FF", "commandPort":8000, "heartbeatPort":6789, "udpPort":10000 }

BODY参数定义：

ipaddr	IP地址
submask	子网掩码
gateway	网关地址
macaddr	MAC地址
commandPort	指令端口
heartbeatPort	心跳端口
udpPort	UDP端口

(2) 镜头参数

URL	http://< ip >:< port >/Config/lens?Camera=< param >
URL参数: Camera=1:左目可见光 Camera=2:右目可见光 Camera=3:TOF	
METHOD	GET
BODY	<pre>{ "focal_length_x":0, "focal_length_y":0, "optical_center_point_x":0, "optical_center_point_y":0, "radia_distortion_coef_k1":0, "radia_distortion_coef_k2":0, "tangential_distortion_p1":0, "tangential_distortion_p2":0 }</pre>
注:	单目版本只有1和3，双目无TOF版本只有1和2

BODY参数定义： (float)

focal_length_x	焦距长度x (像素)
focal_length_y	焦距长度y (像素)
optical_center_point_x	光心投影坐标x
optical_center_point_y	光心投影坐标y
radia_distortion_coef_k1	radtan相机畸变模型畸变系数k1
radia_distortion_coef_k2	radtan相机畸变模型畸变系数k2
tangential_distortion_p1	radtan相机畸变模型畸变系数p1
tangential_distortion_p2	radtan相机畸变模型畸变系数p2

(3) IMU内参

URL	http://< ip >:< port >/Config/imuInter
METHOD	GET
BODY	{ "acc_n":0, "acc_w":0, "gyr_n":0, "gyr_w":0 }

BODY参数定义：（float）

acc_n	加速度计噪声
acc_w	加速度计随机游走
gyr_n	陀螺仪噪声
gyr_w	陀螺仪随机游走

(4) smart参数

URL	http://< ip >:< port >/Config/smart
METHOD	GET/PUT
BODY	{ "gray_image_enable": 0, "imu_enable": 0, "tof_enable": 0, "tof_deep_image_enable": 0, "tof_amp_image_enable": 0 }

BODY参数定义：

gray_image_enable	灰度图启用： 3 / 2 / 1 / 0
imu_enable	Imu启用： 1/0
tof_enable	Tof启用： 1/0
tof_deep_image_enable	Tof深度图启用： 1/0
tof_amp_image_enable	Tof幅度图启用： 1/0

注：灰度图启用：

- 0：不启用流获取灰度图
- 1：启用流获取左目灰度图（单目）
- 2：启用流获取右目灰度图
- 3：启用流获取双目灰度图

(5) 重定位

URL	http://< ip >:< port >/Smart/relocation
METHOD	PUT
BODY	[0,0,0,0,0,0,0,0,0,0,0,0]

BODY参数定义：一个3行4列的位姿变换矩阵，由12个浮点数组成的数组，每4个值表示矩阵的一行

(6) Vio算法控制

1) Vio算法启用

URL	http://< ip >:< port >/Algorithm/enable/"algo_tyep_num"
METHOD	PUT
BODY	无

URL	"algo_tyep_num": http://<ip>:<port>/Algorithm/enable/"algo_tyep_num"
注:	1.stereo1 2.stereo2 3.mono

2) Vio算法禁用

URL	http://<ip>:<port>/Algorithm/disable/"algo_tyep_num"
METHOD	PUT
BODY	无
注:	"algo_tyep_num": 1.stereo1 2.stereo2 3.mono

3) Vio算法重启

URL	http://<ip>:<port>/Algorithm/reboot/"algo_tyep_num"
METHOD	PUT
BODY	无
注:	"algo_tyep_num": 1.stereo1 2.stereo2 3.mono

4) Vio算法重置

URL	http://<ip>:<port>/Algorithm/reset/"algo_tyep_num"
METHOD	PUT
BODY	无
注:	"algo_tyep_num": 1.stereo1 2.stereo2 3.mono

(7) 回环添加关键帧

URL	http://<ip>:<port>/Smart/addKeyFrame
METHOD	PUT
BODY	无

(8) 回环保存关键帧

URL	http://< ip >:< port >/Smart/saveKeyFrame
METHOD	PUT
BODY	无

(9) cam2imu参数

URL	http://< ip >:< port >/Config/cam2imu
METHOD	GET
BODY	[0,0,0,0, 0,0,0,0, 0,0,0,0]

BODY参数定义：一个3行4列的变换矩阵，由12个浮点数组成的数组，每4个值表示矩阵的一行

(10) tof2imu参数

URL	http://< ip >:< port >/Config/tof2imu
METHOD	GET
BODY	[0,0,0,0, 0,0,0,0, 0,0,0,0]

BODY参数定义：一个3行4列的变换矩阵，由12个浮点数组成的数组，每4个值表示矩阵的一行

(11) 获取数据流

URL	http://< ip >:< port >/Stream?Channel=< chan >
	chan:数据流通道号 通道1: imu + stereo1位姿 + stereo2位姿 + 系统状态 通道2: 左目可见光灰度图 通道3: 深度图 + 幅度图 通道4: 算法输出点云 通道5: tof点云 通道6: 右目可见光灰度图
METHOD	GET
BODY	无
注:	单目版本可见光图为通道2: 左目可见光, 位姿为stereo1位姿

数据包=帧头+帧数据;

帧头=0x33cccc33+帧类型（uint）+时间戳（uint）+序列号（uint）+宽（uint）+高（uint）+长度（uint）；

服务端收到数据流请求后开始发送连续数据包。

当服务端收到Bye，则停止发送数据流，并断开连接。

帧类型编号	帧类型
1	IMU
2	左目可见光灰度图
3	深度图
4	幅度图
5	stereo1位姿
6	stereo2位姿
7	stereo2算法点云
8	系统状态
9	tof点云
10	右目可见光灰度图