



celepixel

芯仑科技
CeleX™ 相机套件
IMUProc SDK 使用手册

芯仑科技（上海）有限公司

目录

修改记录.....	3
1 CeleX™ Sensor IMUProc 模块的基本介绍.....	4
1.1 CeleX™ Sensor IMU 硬件模组简介	4
1.2 CeleX™ Sensor IMU 硬件模组输出简介	4
1.3 CeleX™ Sensor IMUProc 模块输出简介.....	4
2 API 描述	6
2.1 概述.....	6
2.2 IMUProc Class Reference	6
2.2.1 updateTiltPanRoll6Axis	7
2.2.2 updateTiltPanRoll9Axis	7
2.2.3 setInitTilt	8
2.2.4 setInitPan.....	8
2.2.5 setInitRoll.....	9
2.2.6 setAutoUpdateStatus	9
2.2.7 setAutoUpdateThreshold.....	9
2.2.8 setXBiasGyro	10
2.2.9 setYBiasGyro	10
2.2.10 setZBiasGyro.....	10
2.2.11 setAutoUpdateThreshold.....	11
2.2.12 configIMUCalib	11
2.2.13 getIMUData.....	11
2.2.14 getInitializeStatus	12
2.2.15 getAutoUpdateStatus.....	12
2.2.16 getXBiasGyro.....	13
2.2.17 getYBiasGyro.....	14
2.2.18 getZBiasGyro.....	14

修改记录

Version	Date	Section	Description	Author
1.0	2018.09.27	All	New	Yueyin Zhou
1.1	2018.10.18	2.2	1. Add 9 axis fusion function 2. Add imu calibration configuration function 3. Add real car support functions 4. Modify tab.1-1 contents	Yueyin Zhou
1.2	2018.10.26	1.3, 2.2	Add output structure IMUData and the corresponding function	Yueyin Zhou
1.3	2018.11.02	1.3, 2.2	1. Delete unnecessary get functions 2. add pan_rate output in IMUData	Yueyin Zhou
1.4	2019.04.30	1.3,2.2	1. add delta_{tilt,pan,roll} output 2. add setAppMode function 3. add getCompensatedGyro/ getCompensatedAcc/ getCompensatedMag functions	Yueyin Zhou

1 CeleX™ Sensor IMUProc 模块的基本介绍

1.1 CeleX™ Sensor IMU 硬件模组简介

CeleX™ Sensor IMU 硬件使用的是 MPU-9250 模组, 是集成 9 轴姿态数据输出的传感器, 安装在 CeleX™ Sensor 背部中心位置。

关于 MPU-9250 详细信息, 请参考官网 datasheet, 地址如下:

<https://www.digikey.ie/en/resources/datasheets/invensense/mpu-9250-product-spec>

1.2 CeleX™ Sensor IMU 硬件模组输出简介

CeleX™ Sensor IMU 原始输出测量数据主要包含三轴陀螺仪数据, 三轴加速度数据, 以及三轴磁力计数据, 具体见表 1-1:

表 1-1 MPU-9250 原始输出测量数据

Group	Terminology	Description
Gyro	gyro_x	绕 x 轴方向旋转角速度, 单位 rad/s, 默认量程 $\pm 250^{\circ}/s$, 对应 tilt 角
	gyro_y	绕 y 轴方向旋转角速度, 单位 rad/s, 默认量程 $\pm 250^{\circ}/s$, 对应 pan 角
	gyro_z	绕 z 轴方向旋转角速度, 单位 rad/s, 默认量程 $\pm 250^{\circ}/s$, 对应 roll 角
Acc	acc_x	绕 x 轴方向加速度, 单位 m/s^2 , 默认量程 $\pm 2g$
	acc_y	绕 y 轴方向加速度, 单位 m/s^2 , 默认量程 $\pm 2g$
	acc_z	绕 z 轴方向加速度, 单位 m/s^2 , 默认量程 $\pm 2g$
Mag	mag_x	绕 x 轴方向磁力计输出, 单位 μT , 默认量程 $\pm 4800 \mu T$
	mag_y	绕 y 轴方向磁力计输出, 单位 μT , 默认量程 $\pm 4800 \mu T$
	mag_z	绕 z 轴方向磁力计输出, 单位 μT , 默认量程 $\pm 4800 \mu T$

1.3 CeleX™ Sensor IMUProc 模块输出简介

基于章节 1.2 中的内容, 以及精确的时间戳数据(单位 ms), 本模块将融合输出对应时间戳下的三个姿态角数据, 分别为俯仰角 tilt、航向角 pan 和横滚角 roll, 以及其他相关数据, 这些数据存出于 `IMUData` 结构体中, 使用时需要指明环境变量 `imu_proc`, 具体详见表 1-2, 图 1-1。

表 1-2 IMUProc 模块输出数据定义

No	Terminology	Description
1	cur_tilt	当前俯仰角, 水平面向上为正, 范围 $\pm \pi/2$, 单位 rad
2	cur_pan	当前航向角, 投影到水平面上逆时针为正, 范围 $\pm \pi$, 单位 rad
3	cur_roll	当前横滚角, 沿相机光轴向前, 逆时针为正, 范围 $\pm \pi/2$, 单位

		rad
4	dynamic_tilt_base	动态自调整的参考俯仰角基准, 适用于实车数据基准参考
5	dynamic_pan_base	动态自调整的参考航向角基准, 适用于实车数据基准参考
6	dynamic_roll_base	动态自调整的参考横滚角基准, 适用于实车数据基准参考
7	init_tilt	静态固定初始基准俯仰角, 适用于手持数据基准参考
8	init_pan	静态固定初始基准航向角, 适用于手持数据基准参考
9	init_roll	静态固定初始基准横滚角, 适用于手持数据基准参考
10	delta_tilt	获取用于动态图像补偿的俯仰角偏差
11	delta_pan	获取用于动态图像补偿的航向角偏差
12	delta_roll	获取用于动态图像补偿的横滚角偏差
13	acc_lon	相机纵向加速度在水平面的投影, 向前为正
14	acc_lat	相机横向加速度在水平面的投影, 向右为正
15	acc_ground	相机沿垂线指向地面的加速度, 向地面为正
16	cur_pan_rate	航向角变化率, 单位 rad/s

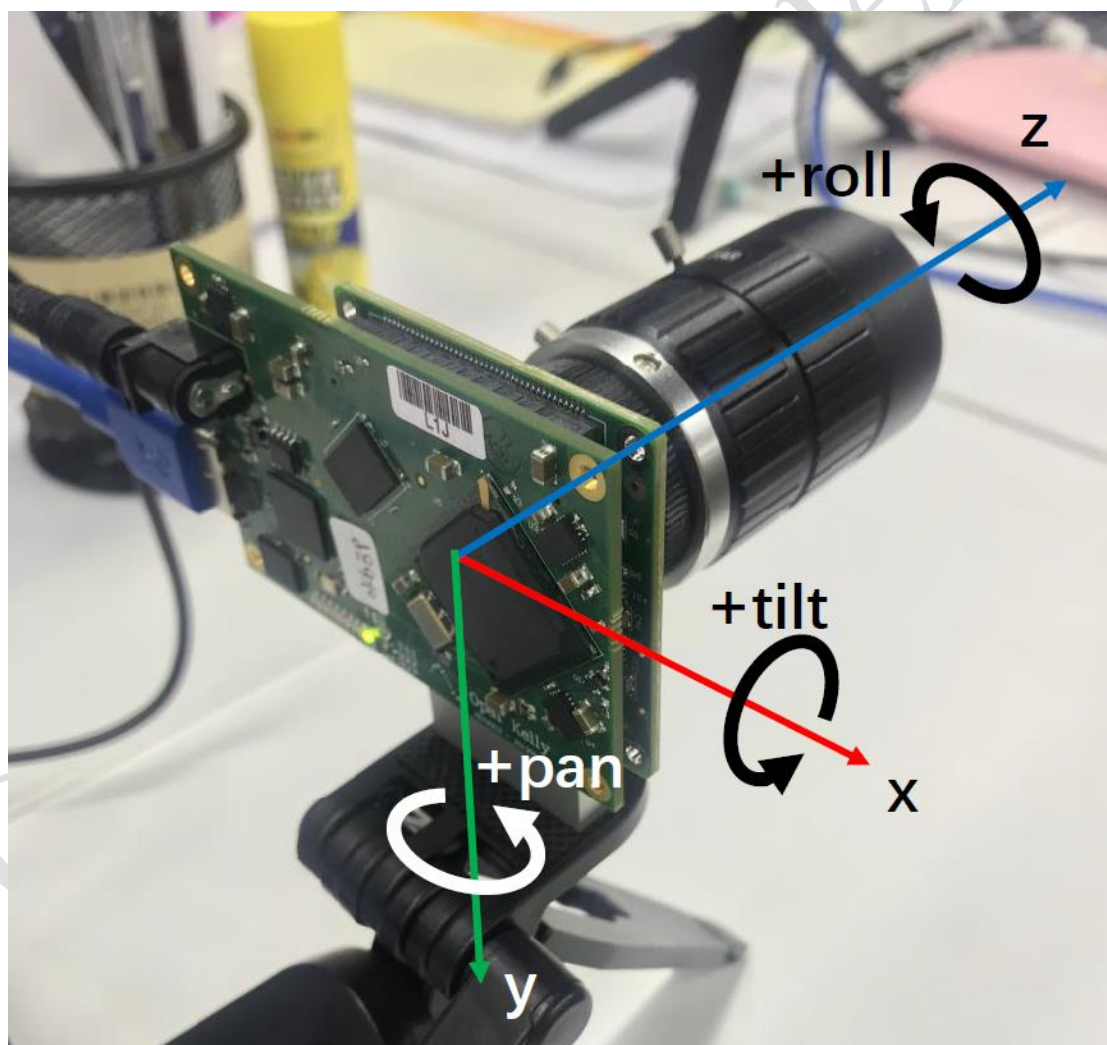


图 1-1 IMUProc 模块输出姿态角示意

2 API 描述

2.1 概述

CeleX™ Sensor IMUProc 模块接口为 C++ 接口。要使用此 API 构建应用程序，您需要将 include 目录中的文件拷贝到您的项目中，这些头文件中包含了我们提供的所有接口。同时，还需指定 IMUProc.dll 和 IMUProc.lib 的路径。

此外，该模块依赖 iniParser 模块，用于读取解析 IMU 标定参数。

2.2 IMUProc Class Reference

对外开放可供调用的核心函数功能，就是将带当前时间戳的原始九轴测量数据，通过滤波融合算法，解算出对外可用且稳定的三个姿态角数据。除此之外，还包括一些初始角度配置、陀螺仪偏差设置、获取当前姿态角数据等功能。

Public Member Functions:

API Name	Description
updateTiltPanRoll6Axis	输入当前时间戳下的 6 轴原始观测值
updateTiltPanRoll9Axis	输入当前时间戳下的 9 轴原始观测值
setInitTilt	设置初始俯仰角，默认为 0.0，单位 rad
setInitPan	设置初始航向角，默认为 0.0，单位 rad
setInitRoll	设置初始横滚角，默认为 0.0，单位 rad
setAutoUpdateStatus	设置自动更新初始角度模式，默认为 false(不自动更新)，该功能打开，则当姿态角变化量大于给定阈值后，重新设置当前姿态角为初始姿态角。
setAutoUpdateThreshold	设置自动更新角度阈值，仅设置 tilt 和 pan，单位 rad
setXBiasGyro	设置陀螺仪 x 轴方向的固定 bias，单位 rad/s，建议使用 configIMUCalib
setYBiasGyro	设置陀螺仪 y 轴方向的固定 bias，单位 rad/s，建议使用 configIMUCalib
setZBiasGyro	设置陀螺仪 z 轴方向的固定 bias，单位 rad/s，建议使用 configIMUCalib
configIMUCalib	一步设置 IMU 所有标定参数，输入 INIParser 对象的指针
setAppMode	设置当前应用模式，分手持模式 HAND_HOLD 和实车模式 REAL_CAR
getIMUData	一步获取 IMUProc 当前所有数据信息，存于 imu_proc::IMUData 结构体中
getInitializeStatus	获取当前初始化状态，true 表示初始化完毕
getAutoUpdateStatus	获取当前自动更新初始角度模式，true 表示打开
getCompensatedGyro	根据配置的标定参数，获取参数补偿后的三轴陀螺仪数据
getCompensatedAcc	根据配置的标定参数，获取参数补偿后的三轴加速度计数据
getCompensatedMag	根据配置的标定参数，获取参数补偿后的三轴磁力计数据
getXBiasGyro	获取当前陀螺仪 x 轴方向的固定 bias，单位 rad/s
getYBiasGyro	获取当前陀螺仪 y 轴方向的固定 bias，单位 rad/s
getZBiasGyro	获取当前陀螺仪 z 轴方向的固定 bias，单位 rad/s

2.2.1 updateTiltPanRoll6Axis

```
int updateTiltPanRoll6Axis (  
    const uint64_t timestamp,  
    const double acc_x,  
    const double acc_y,  
    const double acc_z,  
    const double gyro_x,  
    const double gyro_y,  
    const double gyro_z);
```

Parameters

[in] timestamp	当前时间戳，单位 ms；
[in] acc_x	加速度计 x 轴数据，单位 m/s^2 ；
[in] acc_y	加速度计 y 轴数据，单位 m/s^2 ；
[in] acc_z	加速度计 z 轴数据，单位 m/s^2 ；
[in] gyro_x	陀螺仪 x 轴角速度，单位 rad/s ；
[in] gyro_y	陀螺仪 y 轴角速度，单位 rad/s ；
[in] gyro_z	陀螺仪 z 轴角速度，单位 rad/s ；

该方法执行融合解算过程，融合不包括磁力计的 6 轴原始输出数据，解算结果通过对应的 get 方法，得到对应的姿态角数据。

Return:

ErrorCode

See also

setInitTilt, setInitPan, setInitRoll

2.2.2 updateTiltPanRoll9Axis

```
int updateTiltPanRoll9Axis (  
    const uint64_t timestamp,  
    const double acc_x,  
    const double acc_y,  
    const double acc_z,  
    const double gyro_x,  
    const double gyro_y,  
    const double gyro_z,  
  
    const double mag_x,  
    const double mag_y,
```

```
const double mag_z);
```

Parameters

[in] timestamp	当前时间戳, 单位 ms;
[in] acc_x	加速度计 x 轴数据, 单位 m/s^2 ;
[in] acc_y	加速度计 y 轴数据, 单位 m/s^2 ;
[in] acc_z	加速度计 z 轴数据, 单位 m/s^2 ;
[in] gyro_x	陀螺仪 x 轴角速度, 单位 rad/s ;
[in] gyro_y	陀螺仪 y 轴角速度, 单位 rad/s ;
[in] gyro_z	陀螺仪 z 轴角速度, 单位 rad/s ;
[in] mag_x	磁力计 x 轴测量值, 单位 μT ;
[in] mag_y	磁力计 y 轴测量值, 单位 μT ;
[in] mag_z	磁力计 z 轴测量值, 单位 μT ;

该方法执行融合解算过程, 融合所有 9 轴原始输出数据, 解算结果通过对应的 get 方法, 得到对应的姿态角数据。正常输出数据前, 需要大概 5 秒时间, 用于初始化磁力计输出。

Return:

ErrorCode

See also

setInitTilt, setInitPan, setInitRoll

2.2.3 setInitTilt

```
void setInitTilt(const double init_tilt = 0.0);
```

Parameters

[in] init_tilt	需要设置的初始俯仰角, 默认 0.0, 单位 rad;
----------------	-----------------------------

该方法用于设置初始状态的俯仰角, 随时可配置。自动初始化过程结束后, 会将初始化后的当前俯仰角自动设置为初始俯仰角。

See also

setInitPan, setInitRoll

2.2.4 setInitPan

```
void setInitPan(const double init_pan = 0.0);
```

Parameters

[in] `init_pan` 需要设置的初始航向角，默认 0.0，单位 rad;

该方法用于设置初始状态的航向角，随时可配置。自动初始化过程结束后，会将初始航向角归零。

See also

`setInitTilt`, `setInitRoll`

2.2.5 setInitRoll

```
void setInitRoll(const double init_roll = 0.0);
```

Parameters

[in] `init_roll` 需要设置的初始横滚角，默认 0.0，单位 rad;

该方法用于设置初始状态的横滚角，随时可配置。自动初始化过程结束后，会将初始化后的当前横滚角自动设置为初始横滚角。

See also

`setInitTilt`, `setInitPan`

2.2.6 setAutoUpdateStatus

```
void setAutoUpdateStatus(const bool auto_update = false);
```

Parameters

[in] `auto_update` 是否开启自动更新初始姿态角，默认 false，不开启

该方法用于设置是否自动更新初始姿态角，开启后可以保证当前姿态角与初始姿态角之间的偏差量小于给定阈值范围内，当偏差量大于给定阈值时，自动重新设置当前姿态角为初始姿态角。

See also

`getAutoUpdateStatus`, `setAutoUpdateThreshold`

2.2.7 setAutoUpdateThreshold

```
void setAutoUpdateThreshold(const double tilt_threshold = 0.6, const double pan_threshold = 0.6);
```

Parameters

[in] `tilt_threshold` tilt 方向自动重置阈值

[in] `pan_threshold` pan 方向自动重置阈值

该方法用于设置自动重置 tilt 和 pan 的姿态角阈值，当前姿态角与初始姿态角之间超过该阈值时，自动将当前姿态角重置为初始姿态角。

See also

setAutoUpdateStatus, getAutoUpdateStatus

2.2.8 setXBiasGyro

```
void setXBiasGyro(const double gyro_x_bias = 0.0);
```

Parameters

[in] gyro_x_bias 设置当前陀螺仪 x 轴固定偏差

该方法用于设置陀螺仪 x 轴的固定偏差，用于抵消随时间积分零漂，单位 rad/s，可由长时间静止放置的数据，经离线标定得到该参数。

See also

setYBiasGyro, setZBiasGyro, getXBiasGyro, getYBiasGyro, getZBiasGyro, configIMUCalib

2.2.9 setYBiasGyro

```
void setYBiasGyro(const double gyro_y_bias = 0.0);
```

Parameters

[in] gyro_y_bias 设置当前陀螺仪 y 轴固定偏差

该方法用于设置陀螺仪 y 轴的固定偏差，用于抵消随时间积分零漂，单位 rad/s，可由长时间静止放置的数据，经离线标定得到该参数。

See also

setXBiasGyro, setZBiasGyro, getXBiasGyro, getYBiasGyro, getZBiasGyro, configIMUCalib

2.2.10 setZBiasGyro

```
void setZBiasGyro(const double gyro_z_bias = 0.0);
```

Parameters

[in] gyro_z_bias 设置当前陀螺仪 z 轴固定偏差

该方法用于设置陀螺仪 z 轴的固定偏差，用于抵消随时间积分零漂，单位 rad/s，可由长时间静止放置的数据，经离线标定得到该参数。

See also

setXBiasGyro, setYBiasGyro, getXBiasGyro, getYBiasGyro, getZBiasGyro, configIMUCalib

2.2.11 setAutoUpdateThreshold

```
void setAutoUpdateThreshold(const double tilt_threshold = 0.6, const double pan_threshold = 0.6);
```

Parameters

[in] `tilt_threshold` tilt 方向自动重置阈值

[in] `pan_threshold` pan 方向自动重置阈值

该方法用于设置自动重置 tilt 和 pan 的姿态角阈值，当前姿态角与初始姿态角之间超过该阈值时，自动将当前姿态角重置为初始姿态角。

See also

`setAutoUpdateStatus`, `getAutoUpdateStatus`

`setXBiasGyro`

2.2.12 configIMUCalib

```
int configIMUCalib(util::INIParser* ptr_imu_calib_ini_parser);
```

Parameters

[in] `str_calib_file` IMU 标定参数 ini 文件路径

该方法用于一步设置 IMU 的所有内部标定参数，参数值全部存于指定的 ini 文件中 (IMUCalibParams.ini)，由给定标定工具标定自动得到。

Return:

ErrorCode

2.2.13 setAppMode

```
void setAppMode(int app_mode);
```

Parameters

[in] `app_mode` 枚举类应用模式，分 `HAND_HOLD` 和 `REAL_CAR` 两种模式

该方法用于设置 IMU 内部采用不同的融合策略，适用于手持模式或实车模式，并且对于表 1-2 中的“10-12”输出的结果进行统一。

2.2.14 getIMUData

```
void getIMUData(IMUData &imu_data);
```

Parameters

[out] `imu_data` 存储数据的结构体

该方法可以一并输出所有 IMUProc 模块提供的数据信息。

See also

2.2.15 getInitializeStatus

```
bool getInitializeStatus() const;
```

Returns

该方法用于获取当前初始化状态，true 表明初始化完毕。

2.2.16 getAutoUpdateStatus

```
bool getAutoUpdateStatus() const;
```

Returns

该方法用于获取当前自动更新初始姿态角状态，true 表明开启。

See also

setAutoUpdateStatus

2.2.17 getCompensatedGyro

```
void getCompensatedGyro(double src_gyro_x, double src_gyro_y, double src_gyro_z,  
                        double &dst_gyro_x, double &dst_gyro_y, double &dst_gyro_z);
```

Parameters

[in] src_gyro_x	输入的陀螺仪 x 轴原始数据
[in] src_gyro_y	输入的陀螺仪 y 轴原始数据
[in] src_gyro_z	输入的陀螺仪 z 轴原始数据
[out] dst_gyro_x	输出的陀螺仪 x 轴补偿后数据
[out] dst_gyro_y	输出的陀螺仪 y 轴补偿后数据
[out] dst_gyro_z	输出的陀螺仪 z 轴补偿后数据

See also

configIMUCalib

2.2.18 getCompensatedAcc

```
void getCompensatedAcc(double src_acc_x, double src_acc_y, double src_acc_z,  
    double &dst_acc_x, double &dst_acc_y, double &dst_acc_z);
```

Parameters

[in] src_acc_x	输入的加速度计 x 轴原始数据
[in] src_acc_y	输入的加速度计 y 轴原始数据
[in] src_acc_z	输入的加速度计 z 轴原始数据
[out] dst_acc_x	输出的加速度计 x 轴补偿后数据
[out] dst_acc_y	输出的加速度计 y 轴补偿后数据
[out] dst_acc_z	输出的加速度计 z 轴补偿后数据

See also

configIMUCalib

2.2.19 getCompensatedMag

```
void getCompensatedMag(double src_mag_x, double src_mag_y, double src_mag_z,  
    double &dst_mag_x, double &dst_mag_y, double &dst_mag_z);
```

Parameters

[in] src_mag_x	输入的磁力计 x 轴原始数据
[in] src_mag_y	输入的磁力计 y 轴原始数据
[in] src_mag_z	输入的磁力计 z 轴原始数据
[out] dst_mag_x	输出的磁力计 x 轴补偿后数据
[out] dst_mag_y	输出的磁力计 y 轴补偿后数据
[out] dst_mag_z	输出的磁力计 z 轴补偿后数据

See also

configIMUCalib

2.2.20 getXBiasGyro

```
double getXBiasGyro();
```

Returns

该方法用于获取当前设置的陀螺仪 x 轴偏差，单位 rad/s。

See also

setXBiasGyro, setYBiasGyro, setZBiasGyro, getYBiasGyro, getZBiasGyro

2.2.21 getYBiasGyro

```
double getYBiasGyro ();
```

Returns

该方法用于获取当前设置的陀螺仪 y 轴偏差，单位 rad/s。

See also

setXBiasGyro, setYBiasGyro, setZBiasGyro, getXBiasGyro, getZBiasGyro

2.2.22 getZBiasGyro

```
double getZBiasGyro ();
```

Returns

该方法用于获取当前设置的陀螺仪 z 轴偏差，单位 rad/s。

See also

setXBiasGyro, setYBiasGyro, setZBiasGyro, getXBiasGyro, getYBiasGyro