TMTVision SDK

──────── 底层架构2.0Beta

目录

[一、 模块化多线程编程 2](#_Toc451758962)

[二、 使用方法 4](#_Toc451758963)

[三、 进度与改进 5](#_Toc451758964)

# 模块化多线程编程

为了方便搭建多线程程序，实现模块化多线程编程。在原有库的基础上，将设备和数据对象封装为支持读写的Device类；将线程操作封装为MotoLink类；将算法等处理过程封装为Process类。

模块中定义了：

1. 设备类（虚）Device：含Read/operator>>和Write/operator<<函数用于读写设备IO，其析构函数和功能函数均在其派生类中实现。
2. 数据模板类DeData<T>：实现了对数据的深拷贝构造、深拷贝operator=、重复比较operator==；内含数据时间管理，处理标志位，便于异步数据管理。
3. 数据设备类BufferDevice<T>：派生自Device和SafeData，内含数据缓存类CircularBuffer对象；具有多线程保护机制，可以实现占用或非占用的读写缓存。
4. 线程指针MotoLink：派生自Thread，为Device到Device的指针；功能是驱动数据按照指针方向流动：从输入端读入Device数据，写入输出端Device中；具有线程工作频率管理功能。
5. 数据处理类（虚）Processer：派生自设备类Device，输入和输出支持两种以上模板数据类型。

上述内容调用了SDK中的两个基类：

1. 线程管理类Thread：线程类，内置多种中断管理，派生其Task函数实现功能。
2. 环形数据缓存类CircularBuffer<T>：与boost:: circular\_buffer相同，为固定内存和头尾指针实现的环形存储结构，可以实现队列Queue和堆栈Stack功能；支持内存连续的数据对象或含有深拷贝赋值运算符=的数据对象，其他对象需重载其赋值运算符=为深拷贝。
3. 多线程保护类SafeData：含有多种多线程保护机制，可以实现占用或非占用的读写缓存。

继承和调用关系如图所示：



图1 继承和调用关系

# 使用方法

由于BufferDevice等均派生自Device基类，具有相同的IO读写函数接口>>和<<，可以自由组合串联；IO接口内置多线程保护，支持并联模式。

数据类DeData包含重复比较和生命周期管理方法；线程指针MotoLink含工作频率分析；数据设备类BufferDevice含存储空间分析方法，便于分析负载。使用方法如图所示：

|  |  |
| --- | --- |
| a.异步读设备处理并写入缓存 | b. 异步并行读写和处理数据 |
| c.利用网络设备实现多进程数据交互 | |

图2使用方法的多样性

# 进度与改进



图3 进度

改进：设备基类上需要增加回调函数接口，使设备能够主动回调用户方法。



*2016/5/23*

[*smartsupervisor@163.com*](mailto:smartsupervisor@163.com)

[*towanglei@163.com*](mailto:towanglei@163.com)