**系统框架设计方案**

### 需求描述：

C:\Documents and Settings\Administrator\Desktop\Image2.EMF

1. **配置文件解析：** 完成配置文件解析。

从文件中读取参数，解析为key-value格式。保存在std::map<std::string,std::string>类型的map中。

1. **传感器数据缓存：**完成传感器数据的缓存。

将传感器数据根据数据类型和订阅者分别保存在不同的队列中。

1. **传感器参数配置：**完成传感器的参数配置。

1. **启动传感器：**驱动传感器开始收集数据。
2. **传感器数据读取：**将传感器收集的数据保存到缓存中。

1. **同频传感器数据读取：**将一组同频传感器收集的数据尽可能同时的写入到缓存中。

*注：该功能为功能5的一个子集。*

1. **启动算法：**驱动算法模块开始计算。

算法模块从缓存中读取需要的传感器数据，然后进行计算。

1. **数据持久化：**将传感器数据保存到文件中。
2. **线程守护：**监控本系统的线程状态。

可靠性需求，低优先级。

1. **日志：**保存系统运行时的信息。

低优先级。

### **模块划分：**

**Diagram: component**



1. **缓存模块：**

* 提供创建和销毁publisher接口，有数据发布需求的模块，通过publisher的write接口可以发布指定类型的数据，算法模块也可以发布数据。
* 提供创建和销毁subscriber接口，通过subscriber的read接口可以读取指定类型的数据。
* 同一种数据类型的多个subscriber拥有独立的缓存队列，相互独立。
* 缓存队列中的数据报文被读取之后会被删除。
* 如果一种数据没有订阅者，则该类型的数据不会写入缓存。

1. **传感器模块：**

* 根据配置文件创建传感器对象，框架为每个对象创建一个线程，持续从驱动读取传感器数据并写入缓存队列。
* 传感器对象分为两种类型，独立传感器和传感器组。传感器组是一组同频传感器的集合。读取传感器数据或写入缓存会同时操作组内的所有传感器。

1. **算法模块：**

* 根据配置文件创建算法对象。
* 是否启动线程、订阅何种数据由算法内部决定。
* 算法也可以发布指定类型的数据到缓存队列中。

### **静态类图：**

**Diagram: queue**



**Diagram: sensor**



**Diagram: alg**

C:\Documents and Settings\Administrator\Desktop\Image2.EMF

**Diagram: basic data type**

C:\Documents and Settings\Administrator\Desktop\Image2.EMF

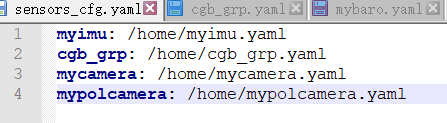
### **配置文件：**

1. **系统配置文件：**

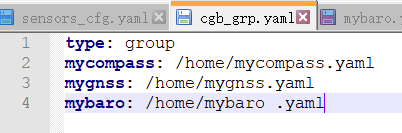
用于配置系统全局参数，比如日志或打印级别。

1. **传感器配置文件**

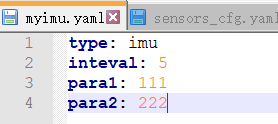
* 每行代表一个传感器对象，包括名称和配置文件。



* type如果是group，表示该对象是一组同频传感器，则后续的每一行代表一个传感器对象。

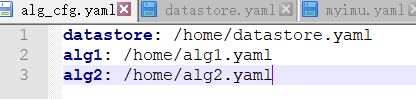


* type如果是具体的传感器，则后续就是具体参数，参数可根据不同的传感器增删。



1. **算法配置文件**

* 每行代表一个算法对象，包括名称和配置文件。



* 配置文件包括算法名称、需要的传感器数据、队列长度、读取数据间隔等，可根据需要增删。

