系统机制细节解读

[1 动作细节 1](#_Toc186195420)

[1.1 位置定标 1](#_Toc186195421)

[1.2 使用位置定标（使用LocationV2） 1](#_Toc186195422)

[1.2.1 使用位置定标的初始化的顺序 1](#_Toc186195423)

[1.2.2 Pipettor 2](#_Toc186195424)

[1.2.3 单臂ArmPipettor 2](#_Toc186195425)

[1.2.4 排枪ArrayPipettor 2](#_Toc186195426)

[1.3 位置相关动作 3](#_Toc186195427)

[1.3.1 单臂吸液，MM根据吸液量计算高度，其他根据adf配置的距底高度计算 3](#_Toc186195428)

[1.3.2 单臂Transfer转液的细节\*\*\* 3](#_Toc186195429)

[2 通信机制 3](#_Toc186195430)

[3 流程机制 4](#_Toc186195431)

[3.1 荧光和温度原始数据记录方式 4](#_Toc186195432)

[4 其他 5](#_Toc186195433)

# 动作细节

## 位置定标

单臂核仅在给核提试剂条定标时，Z坐标计算lane0和lane11 Z的平均值，且是在页面内存计算，定标数据中仅保留计算后的平均值。

## 使用位置定标（使用LocationV2）

### 使用位置定标的初始化的顺序

* + 1. TestBed也是单例类。
    2. 创建TestBed的时候会从E方读取荧光头的MPPC高压配置。
    3. 在TestBed.Initialize方法中，先实例化ArrayPipettor；再实例化ArmPipettor，arrayPipettor作为传参；再实例化MasterMixBay，armpipettor作为传参。
    4. 再实例化Trash。
    5. DisposableTipBay加载x、y坐标。
    6. 把DisposableTipBay、Trash、MasterMixBay赋值给ArmPipettor
    7. 其中DisposableTipBay涉及到了库存管理，数据从数据库查询的。
    8. 其他一些LocationV2中的ConsumableSet、ArmPistonLUT、ScaleConverter、CalibrationData、PushHeatLidProcess也在此加载给TestBed类型字段。PushHeatLidProcess是原来上位机存储压盖的定标位置，现在压盖定标位置已经移至EEProm中。

### Pipettor

public Dictionary<LocationInformation, Dictionary<LocationInformation, int>> ZcoordinateOnPlatform；

ZcoordinateOnPlatform是按Tip分的第一个LocationInformation指的是那个Tip。

第二个LocationInformation指的才是tip的位置；int是Z高度。

### 单臂ArmPipettor

构造函数中把LocationV2的数据加载到类型内部的变量中。包括：

* + 1. 设置各种位置的x的位置；
    2. 计算得到非定标位置的PCRTube X坐标：Location中已经由定标点，计算了剩余first little tube的x，此时再根据配置的little tube的间距，计算得到所有48个little tube的X坐标位置。
    3. 96Tip盒和垃圾桶位置，是实际创建Command时候决定的。

### 排枪ArrayPipettor

## 位置相关动作

### 单臂吸液，MM根据吸液量计算高度，其他根据adf配置的距底高度计算

MM根据吸液量和Tube物理模型转化高度。

吸液量是传给活塞电机参数，使电机正向运动实现。

### 单臂Transfer转液的细节\*\*\*

# 通信机制

## AtomicCommand

### Events（message events & command events）

Message级别：OnResult；OnError；OnTimeout；OnMessageCompleted。

AtomicCommand级别：OnSuccess；OnFailed；OnCompleted；OnCompleteFailed。

#### **Message级别**

1. OnTimeout：response=null时
2. OnError：倒数第二位错误码非〇时，注意，存在重写Resrve方法把错误码强制修改的操作。目前仅在温度查询不到温的情况下会这么做，参考通信协议。
3. OnResult：payloadLength>1时，1是指倒数第二字节用于存储错误码。
4. OnMessageCompleted：无论如何都会进入此事件。

#### **AtomicCommand级别**

CommandResult（AtomicCommand.Result）: 每条message有响应，且错误码都非〇时为true。

CommandDataValid（AtomicCommand.DataValid）:默认为true，可在OnResult或OnMessageCompleted逻辑里里判定它true/false。

1. OnSuccess：CommandDataValid=true时触发。
2. OnFailed：CommandDataValid=false时触发。
3. OnCompleted：command.Run方法结束前，最后，CommandResult=true时触发。切实在设置Result的setter里触发的。出于什么考虑未知。
4. OnCompleteFailed：CommandResult=false时触发。

## OnMessageCompleted

1. 最初用于“温控曲线”模块。在设定温度之后指令返回之后，读取设定的温度，把数据点加到曲线上。
2. 23年4月份，在排枪取Tip的command里使用它，以获取再哪里取的Tip。
3. 23年8月份，AtomicCommandGetHardwareConfiguration里使用它，用于判断未能成功获取固件版本。且同时配置了OnResult和OnMessageCompleted。
4. 23年，8月份，温度定标也用这个事件来读取温度，这么用其实是不合适的，应该用OnResult才对。如果应该有Result的，但是没有Result，或者Result与预期格式不一致，这里需要有一个验证机制，然后应该触发一个OnResultError的报警是正解。

## RedoRounds

Redo的条件：CommandResult =false；RedoRequired；RedoIntervalMS！=0；redo次数。解释：存在message执行不成功情况，或存在错误码非〇情况时，如果配置了Redo条件，会执行Redo。但是此时，message级别的事件逻辑已经走过一遍了。

## AtomicCommandFast

把几条指令紧密包装在一起下发，避免中间被Scheduler插入其他指令，这样起到fast的作用？

# 流程机制

## 荧光和温度原始数据记录方式

1. 荧光数据和温度数据是通过不同的csv文件记录的。
2. 温度数据记录在一个单独文件中。
3. 荧光数据每个孔位记录一个csv文件。
4. 荧光数据中也有温度数据，但是从温度数据中根据通道、孔位、assayName等7项数据匹配到，然后赋值给荧光文件数据。
5. 温度数据通过查温得到，但是查温时候若返回的数据存在被标记为未到温的情况，则强行把倒数第2状态位置为255（非0即是错误状态）。这也是日志中为什么会出现诸多255报错的原因。此报错在排错时可忽略。
6. 存储csv注册的时command的OnResult事件，未到温，报错，就走报错事件OnError，不走OnResult了。故未到温则不再csv文件夹中记录数据了。

# 其他