

# 一. 设计

## 1. 报文格式:

报文的组成: 命令字~内容~

## 2. 报文的详细设计:

功能描述			
功能序号	1		
名称	扫描参数更新通知(配置文件)		
命令字	0x21		
描述	UI 进程 向 SCAN 进程通知, 扫描参数发生变化.		
方向	UI 进程 ->SCAN 进程		
客户端数据帧描述			
字节序号	功能描述	数据类型	描述
BYTE 0	命令字	Unsigned char	该数据包命令字 0x21
BYTE 1	配置文件路径	Char[]	配置文件的路径

功能描述			
功能序号	2		
名称	校准		
命令字	0x22		
描述	UI 进程 通知 SCAN 进行校准.		
方向	UI 进程 ->SCAN 进程		
客户端数据帧描述			
字节序号	功能描述	数据类型	描述
BYTE 0	命令字	Unsigned char	该数据包的命令字 0x22

功能描述			
功能序号	3		
名称	模板		
命令字	0x23		
描述	UI 进程 通知 SCAN 进程 模板路径所在.		
方向	UI 进程 ->SCAN 进程		
客户端数据帧描述			
字节序号	功能描述	数据类型	描述

BYTE 0	命令字	Unsigned char	该数据包的命令字 0x23
BYTE 1	模板路径	Char[]	模板文件所在路径

功能描述			
功能序号	4		
名称	共享内存指定数据块的回收		
命令字	0x24		
描述	UI 进程 将 处理完后的数据块通知 SCAN 进程回收.		
方向	UI 进程 ->SCAN 进程		
客户端数据帧描述			
字节序号	功能描述	数据类型	描述
BYTE 0	命令字	Unsigned char	该数据包的命令字
BYTE 1	传递给 UI 共享内存的偏移量数据.	Unsigned int	该数据内容在共享内存中的偏移量

功能描述			
功能序号	5		
名称	图片文件路径通知(非模板扫描)		
命令字	0xA0		
描述	SCAN 进程 向 UI 进程发送 扫描后的文件路径.		
方向	SCAN 进程 -> UI 进程		
客户端数据帧描述			
字节序号	功能描述	数据类型	描述
BYTE 0	命令字	Unsigned char	该数据包的命令字
BYTE 1	文件的路径	Char []	该上传文件的绝对路径

功能描述	
功能序号	6
名称	模板数据块传输(模板扫描)
命令字	0xA1
描述	SCAN 进程 向 UI 进程发送扫描完成后的数据.(模板数据内

		容 :表 1)	
方向		SCAN 进程 -> UI 进程	
客户端数据帧描述			
字节序号	功能描述	数据类型	描述
BYTE 0	命令字	Unsigned char	该数据包的命令字
BYTE 1	共享内存的偏移量	Unsigned int	该数据内容在共享内存中的偏移量

功能描述			
功能序号		7	
名称		校准	
命令字		0xA2	
描述		SCAN 进程 将校准后 结果通知 UI 进程.	
方向		SCAN 进程 -> UI 进程	
客户端数据帧描述			
字节序号	功能描述	数据类型	描述
BYTE 0	命令字	Unsigned char	该数据包的命令字
BYTE 1	校准结果	int	SCAN 进程校准结果

功能描述			
功能序号	8		
名称	SCAN 进程故障描述		
命令字	0xA3		
描述	SCAN 进程 通知 UI 进程故障问题描述.		
方向	SCAN 进程 -> UI 进程		
客户端数据帧描述			
字节序号	功能描述	数据类型	描述
BYTE 0	命令字	Unsigned char	该数据包的命令字
BYTE 1	故障描述	int	故障描述

功能描述	
功能序号	9

名称	SCAN 设置扫描间隔		
命令字	0xA4		
描述	SCAN 设置扫描间隔后，向 UI 进程通知；		
方向	SCAN 进程 -> UI 进程		
客户端数据帧描述			
字节序号	功能描述	数据类型	描述
BYTE 0	命令字	Unsigned char	该数据包的命令字
BYTE 1	故障描述	2 个字节(单位 ms)	扫描间隔时间

## 二. 模板扫描数据描述

//传输的模板数据结构      SCAN --> UI

//该结构的大小: 128 + 400 + 500 + 1 + 1 + 500 = 1530 字节

```
typedef struct _templateScanElement{
```

```
    char        identifyID[128];
    char        objectAnswer[400];
    unsigned char    totalNumberPaper;
    char        toalPaperName[10][50];
    unsigned char    subMapCoordNum;
    CoordsArray    coordsDataToUi;
```

```
}templateScanElement;
```

```
typedef Coords CoordsArray[50];
```

表 1 :

起始字节	字段	数据类型	描述及要求
BYTE 0	考生考号(ID)	Char [128]	该考生的考号 ID, C 字符串. 最大 128 字节
BYTE 128	当前试卷的客观题答案.	Char [400]	当前试卷的客观题答案, 最大 400 字节
BYTE 528	试卷的总份数	Unsigned Char	该考生所有试卷的总份数
BYTE 529	所有试卷的名称	Char [10][50]	试卷所有图片的名称, 最大 10 张图片, 每张图片名称最大 50 个字节

BYTE 1029	切图个数	Unsigned char	试卷的切图个数.
BYTE 1030	切图的所有坐标点数据	Coords CoordsArray[50]	自定义结构体的数组; 包含当前坐标标识,左上角,右下角坐标. 最大 50 个.

坐标组结构(自定义结构体):

//切图坐标点与对应的标识符

//2 \* 5 = 10

```
typedef struct _Coords{
    unsigned short MapID;

    unsigned short TopX;

    unsigned short TopY;

    unsigned short DownX;

    unsigned short DownY;

}Coords;
```

起始字节	字段	数据类型	描述及要求
BYTE 0	坐标组标识	Unsigned short	当前坐标组的标识 ID.
BYTE 2	左上角横坐标点	Unsigned short	坐标组: 左上角横坐标点
BYTE 4	左上角纵坐标点	Unsigned short	坐标组: 左上角纵坐标点
BYTE 6	右下角横坐标点	Unsigned short	坐标组: 右下角横坐标点
BYTE 8	右下角纵坐标点	Unsigned short	坐标组: 右下角纵坐标点

### 三. 程序设计

## 1. IPC 通信键值:

- a) 共享内存: 键值 0x1235
- b) 消息队列: 键值 0x1234

## 2. SCAN 端 IPC 程序设计:

- a) 设计四项业务: 1. 普通扫描数据发送; 2. 模板数据发送; 3. 其他命令的发送; 4. 信息接收;

- b) 程序设计思路:

- 1. 先初始化, 创建共享内存池和消息队列
- 2. 对于普通扫描文件信息, 调用 `create_file_send_filepath()`; 创建该扫描文件, 并发送该文件路径给 UI 进程;
- 3. 对于模板扫描, 需 SCAN 主动创建扫描后的所有文件, 然后调用 `alloc_shmory_block()` 从内存池申请内存, 并获取该地址相对于内存池的偏移量, 然后将模板结构体赋值, 存入该地址; 调用 `Tape_template_sendData()`, 将获得偏移量传入, 发送信息给 UI 进程;
- 4. 对于普通其他信息发送, 只需将该信息的命令, 信息(需遵循规定报文的格式)传入该 `scan_send_msgque()` 方法, 即发送给 UI 进程;
- 5. 调用 `scan_rcv_msgque()` 方法, 接收信息并进行处理;
- 6. 调用 `insert_node()`, 注册命令对应的处理方法; 以使接收到信息后, 有对应的处理方法, 如果接收到的命令无处理方法, 会报错;

7. 调用 `destroy_IPC_QueAndShm()`销毁消息队列和共享内存;

### 3. UI 端 IPC 通信程序设计:

a) 设计两项业务: 1. 接收消息; 2. 发送消息; 3. 释放消息内存

b) 程序设计思路:

1. 先初始化, 获取指定键值的消息队列和共享内存;
2. 发送信息, 调用 `ui_send_fun_mesque()`方法; 以指定的信息格式, 发送消息至 SCAN 进程;
3. 接收信息, 调用 `ui_recv_fun_mesque()`方法, 接收指定格式的信息, 然后进行处理;
4. 对接收的信息, 需要调用 `ui_free_fun_mesque()`, 释放该信息的内存;
5. 调用 `destroy_ipc_Que_Shm()`, 销毁消息队列和共享内存;