历史版本

版本	修改说明	修改人	修改时间
V0.1.0	1.初始版本	吴志涛	2017.03.19
V0.1.1	1.修改缓存逻辑到中间件DLL	吴志涛	2017.03.23
V0.1.2	1.更名为接收器对接手册，并增加函数说明	吴志涛	2017.03.25

目录

附录2 答题过程时序图

1 文档说明

1.1 系统说明

在整个系统中由PC端通过串口发送指令控制接收器，接收器再通过13.56Mhz的天线和2.4G的天线与答题器完成数据交互。

1.2 写作目的

在答题器系统中接收器与DLL之间调用采用**JSON**格式来封装数据，本文讲解DLL与接收器之间的JSON格式的交互指令。

1.3 术语与缩写解释

白名单：已经绑定答题器的列表。
uID ：答题器与接收器的的设备ID。
绑定：答题器要与接收器通讯需要首先绑定一下，在绑定的过程中答题器与接收器会完成**设备ID（uID）**的交换，这个答题器与接收器才能相互识别。
信道：答题器与接收器通讯的的频段。
发送功率：答题器与接收器通讯的2.4G无线信号的发送功率。

2 指令函数说明

2.1 指令 函数集

答题器的 指令函数大致分为3类，答题相关的只需熟悉**绑定**和**答题**相关函数，**设置**相关的函数是在安装调试设备时需要关心的。

- 绑定相关函数

```
// 清除白名单
int clear_wl();
// 开始绑定
int bind_start();
// 停止绑定
int bind_stop();
```

- 答题相关函数

```
// 开始答题
int answer_start(int total, char *answer_str);
// 获取答案
char *get_answer_list();
// 停止作答
int answer_stop();
```

- 设置相关函数

```
// 写入学号
char *set_student_id(char *student_id_str);
// 设置信道
int set_channel(int tx_ch, int rx_ch);
// 设置发送功率
int set_tx_power(int tx_power);
// 查询设备信息
char *get_device_info();
```

2.2 答题器绑定相关指令

2.2.1 清除白名单指令

【对应API函数声明】：

```
int clear_wl();
```

【指令功能简介】：
此函数是清除接收器设备的白名单。
【DLL->接收器指令】：

```
{
  'fun': 'clear_wl'
}
```

【接收器->DLL指令】：

```
{
    'fun': 'clear_wl'
    'result':    '0'
}
// 0 : 成功
// 1 : 失败
```

2.2.2 开启绑定指令

【对应API函数声明】：

```
int bind_start();
```

【指令功能简介】：
此函数的功能是开启接收的绑定功能，绑定的方法：将答题器靠近接收器的刷卡区刷卡，如果听到蜂鸣器叫一下则表示刷卡绑定成功。
注意：

1. 在**V0.1.0的协议版本**中一个接收器最多可以绑定**120**个答题器。

2. 开启一次绑定可以完成对所有答题器的绑定。
3. 绑定完成之后需要，需要停止绑定指令才能停止。

【DLL->接收器指令】：

```
{
  'fun': 'bind_start'
}
```

【接收器->DLL指令】：

```
{
    'fun': 'bind_start'
    'result':    '0'
}
// 0 : 成功
// 1 : 失败
```

2.2.3 停止绑定指令

【对应API函数声明】：

```
int bind_stop();
```

【指令功能简介】：
此函数的功能是停止接收器的绑定功能。
【DLL->接收器指令】：

```
{
  'fun': 'bind_stop'
}
```

【接收器->DLL指令】：

```
{
    'fun': 'bind_stop'
    'result':    '0'
}
// 0 : 成功
// 1 : 失败
```

2.3 答题器作答相关指令

- 普通工作模式（推荐用法）
  这种方式每次作答时都需要重新发送题目，发送题目之后接收器一致处于监听状态，答题器之后才能提交答案，直到接收器收到停止作答指令，接收器将拒绝接受答题器指令。如果要开始新的答题，需要重新发送题目。
- 简单工作模式

这种工作方式比较简单，只用发送一次题目，上位机不会发送停止作答指令，这个样接收器一致处于监听状态，答题器之后随时都能提交答案，此时相当答题器与接收器没有交互过程，仅仅是由答题器发送答案到接收器。这个有上位机软件根据**答题器提交答案的时间**判断答案是对应的那个题目。

注意：普通工作方式可以切换题目，有下发停止作答指令，简单工作方式不发送停止作答指令，用发送停止作答指令。

2.3.1 发送题目指令

【对应API函数声明】：

```
1.  int  answer_start(int total, char *answer_str);
```

【指令功能简介】：
将题目信息发送到接收器，再广播给答题器。
【DLL->接收器指令】：
下面以一个例子来说明题目的JSON格式：
例如下发送4道题目的参数如下:

题目	类型	类型编码	题号	题号编码	作答范围	作答范围编码
第1题	单选	s	1	1	A~D	A-D
第2题	多选	m	13	13	A~F	A-F
第3题	判断	j	24	24	对或错	
第4题	评分	d	27	27	1~5	1-5

对应的JSON格式的数据如下：

```
{
  'fun': 'answer_start',
  'time': '2017-02-15:17:41:07:137',
  'total': '4',
  'questions': [
    {
      'type': 's',
      'id': '1',
      'range': 'A-D'
    },
    {
      'type': 'm',
      'id': '13',
      'range': 'A-F'
    },
    {
      'type': 'j',
      'id': '24',
      'range': ''
    },
    {
      'type': 'd',
      'id': '27',
      'range': '1-5'
    }
  ]
}
```

【接收器->DLL指令】：

```
{
    'fun': 'answer_start'
    'result':    '0'
}
// 0 : 成功
// 1 : 失败
```

2.3.2 获取答案指令

【对应API函数声明】：

```
char * get_answer_list();
```

【指令功能简介】：

获取答题器提交的答案，上位机发送完题目之后，通过轮询的方式来获取答题器提交的答案。

【接收器->DLL指令】：

注意：由于接受器内部RAM空间的限制，无法存放太多的数据，一旦有答题器发送数据上来，接收器会立即发送给上位机。

```
{
  'fun': 'update_answer_list'
  'card_id': 'xxxxxxxxx1',
  'update_time': '2017-02-15:17:41:07:237',
  'total': '4',
  'answers': [
    {
      'type': 's',
      'id': '1',
      'answer': 'A'
    },
    {
      'type': 'm',
      'id': '13',
      'answer': 'BC'
    },
    {
      'type': 'j',
      'id': '24',
      'answer': 'false'
    },
    {
      'type': 'd',
      'id': '27',
      'answer': '3'
    }
  ]
}
```

- card_id: 表示答题器的设备ID。
- update_time: 表示答题器提交答案的时间。
- question_number：答案中的题目数。
- answers: 详细的答案信息

> type: 题目类型(question type)。
>
> > s ：表示单选题
> > m：表示多选题
> > j：表示判断题
> > d：表示数字评分
>
> id: 题号(question id):**题号范围1~99**。
> answer: 当前题目提交的答案。

【DLL->接收器指令】：

```
{
    'fun': 'update_answer_list'
    'result':    '0'
}
// 0 : 成功
// 1 : 失败
```

2.3.3 停止作答指令

【对应API函数声明】：

```
int answer_stop();
```

【指令功能简介】：
此函数的功能是停止作答功能，调用该函数将停止接收器的接收功能，并清除答题器上面显示的题目信息。
【DLL->接收器指令】：

```
{
  'fun': 'answer_stop'
}
```

【接收器->DLL指令】：

```
{
    'fun': 'answer_stop'
    'result':    '0'
}
// 0 : 成功
// 1 : 失败
```

---

4 设置指令

4.1 理论说明

设置指令是配置答题器参数的指令，主要操作是答题器上面的一块标签芯片，设置参数就是往这块标签中写入数据，答题器开机的时候会读取这个里面的设这参数，来调整自己的通讯参数。

4.2 答题器设置学号相关指令

4.2.1 设置学号指令

【对应API函数声明】：

```
char *set_student_id(char *student_id_str);
```

【指令功能简介】：
此函数完成将学生的学号信息写入到答题器。我们给学号分配的空间是10个byte。如果是字节，这个学号字符串需是纯数字，长度最大20位。
【DLL->接收器指令】：

```
{
  'fun': 'set_student_id',
  'student_id': 'xxxxxxxxx'
}
```

【接收器->DLL指令】：

```
{
    'fun': 'set_student_id',
    'card_id': 'xxxxxxxxx',
    'student_id': 'xxxxxxxxx1'
}
```

- card_id：写入学号信息的答题器的uID。
- student_id：写入之后读回的学生ID，10进制学生ID的字符串，长度最大20位。

说明:答题器的 开启绑定 与 写入学号 的关系：

- **写入学号**与**开启绑定**都是占用13.56Mhz的刷卡的硬件资源，所以无法在开启绑定的情况下执行写入学号指令，系统只允许在某一时刻只有一条指令占用此硬件。
- 如果使用接收器给**没有绑定**自己的答题器写入学号，则在写入学号的过程中**自动完成**绑定。
- 如果在本机上已经绑定的答题器，写入学号，则**原绑定信息不变**。

4.3 答题器设置通讯参数相关指令

答题器在与接收器通讯的过程中有一些关键参数有可能需要修改，比如信道和发送功率等，设置的这些参数对通讯是否成功至关重要。
注意：
1. 这些参数是在答题器贴近接收器刷卡区的过程中写入到答题器的。
2. 修改这些参数必须完成刷卡才能生效，建议在完成绑定之前先设定这些参数。
3. 系统通讯关键参数出厂时已经写入了默认值，不修改可以直接使用。

4.3.1 设置信道指令

【对应API函数声明】：

```
int set_channel(int tx_ch,int rx_ch);
```

【指令功能简介】：
这条函数是用来设置答题器与接收的通选信道的，有的时候为了避免干扰，需要修改答题器与接收器的信道，**发送与接收的信道是不能相同。**

- tx_ch：答题器的发送信道，范围1~11。
- rx_ch：答题器的接收信道，范围1~11。

【DLL->接收器指令】：

```
{
  'fun': 'set_channel',
  'tx_ch': '2',
  'rx_ch': '6'
}
```

【接收器->DLL指令】：

```
{
    'fun': 'set_channel',
    'result':    '0'
}
```

4.3.2 设置发送功率指令

【对应API函数声明】：

```
int set_tx_power(int tx_power);
```

【指令功能简介】：
此函数是用来设置答题器的发送功率，我们答题器默认是最大功率发送的，有时候这个功率是比较大的，当应该场景不需要这么大的功率时，我们可以使用这个接口修改发送功率。答题器的发送功率，范围1~5，分别表示对应的不同档位大发送功率，1当最小，5档最大。
【DLL->接收器指令】：

```
{
  'fun': 'set_tx_power',
  'tx_power': '4'
}
```

【接收器->DLL指令】：

```
{
    'fun': 'set_tx_power',
    'result':    '0'
}
```

4.4 其他指令

4.4.1 查询设备信息指令

【对应API函数声明】：

```
char *get_device_info();
```

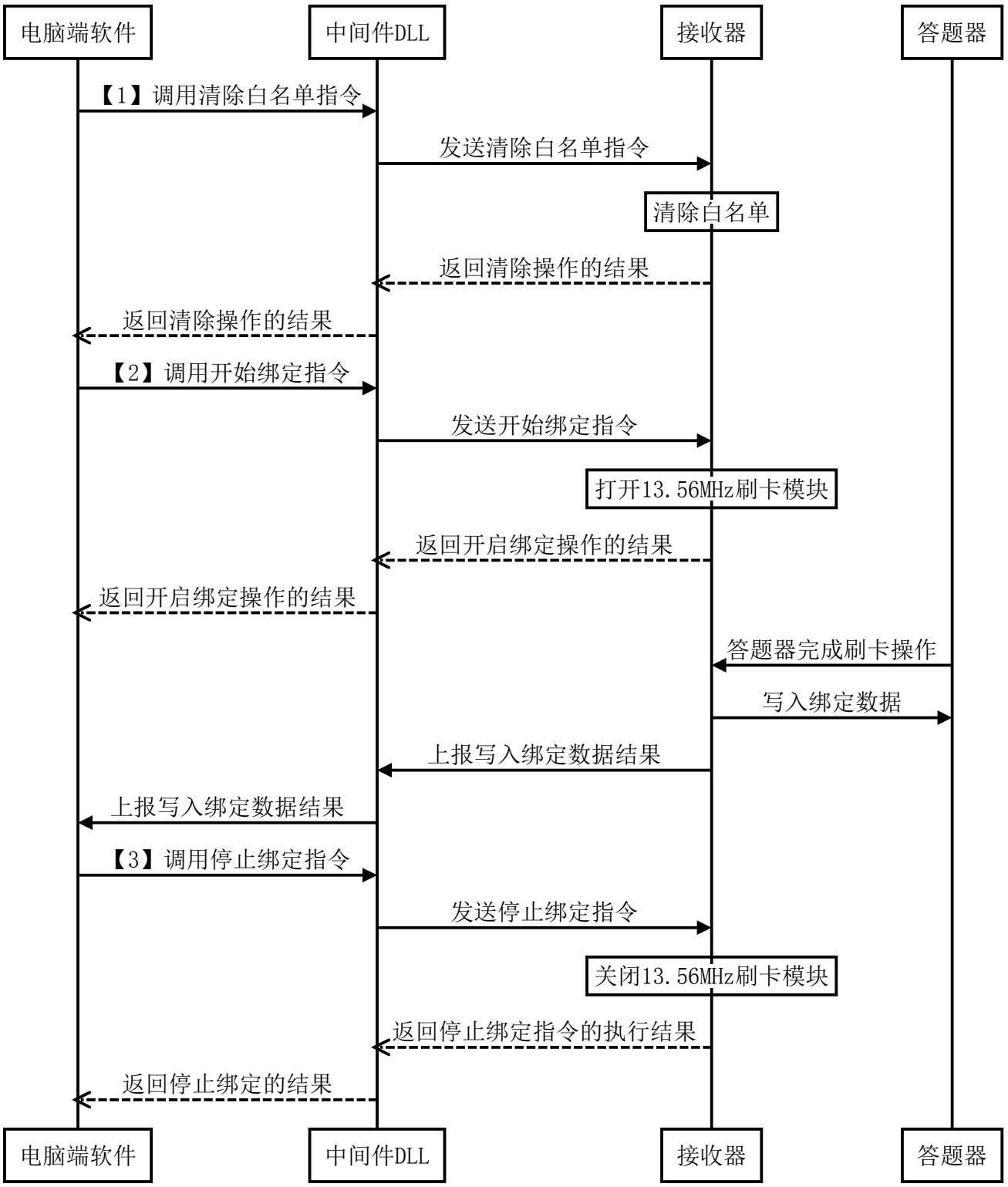
【指令功能简介】：
此函数用来查询接收器的设备信息。
【DLL->接收器指令】：

```
{
  'fun': 'get_device_info'
}
```

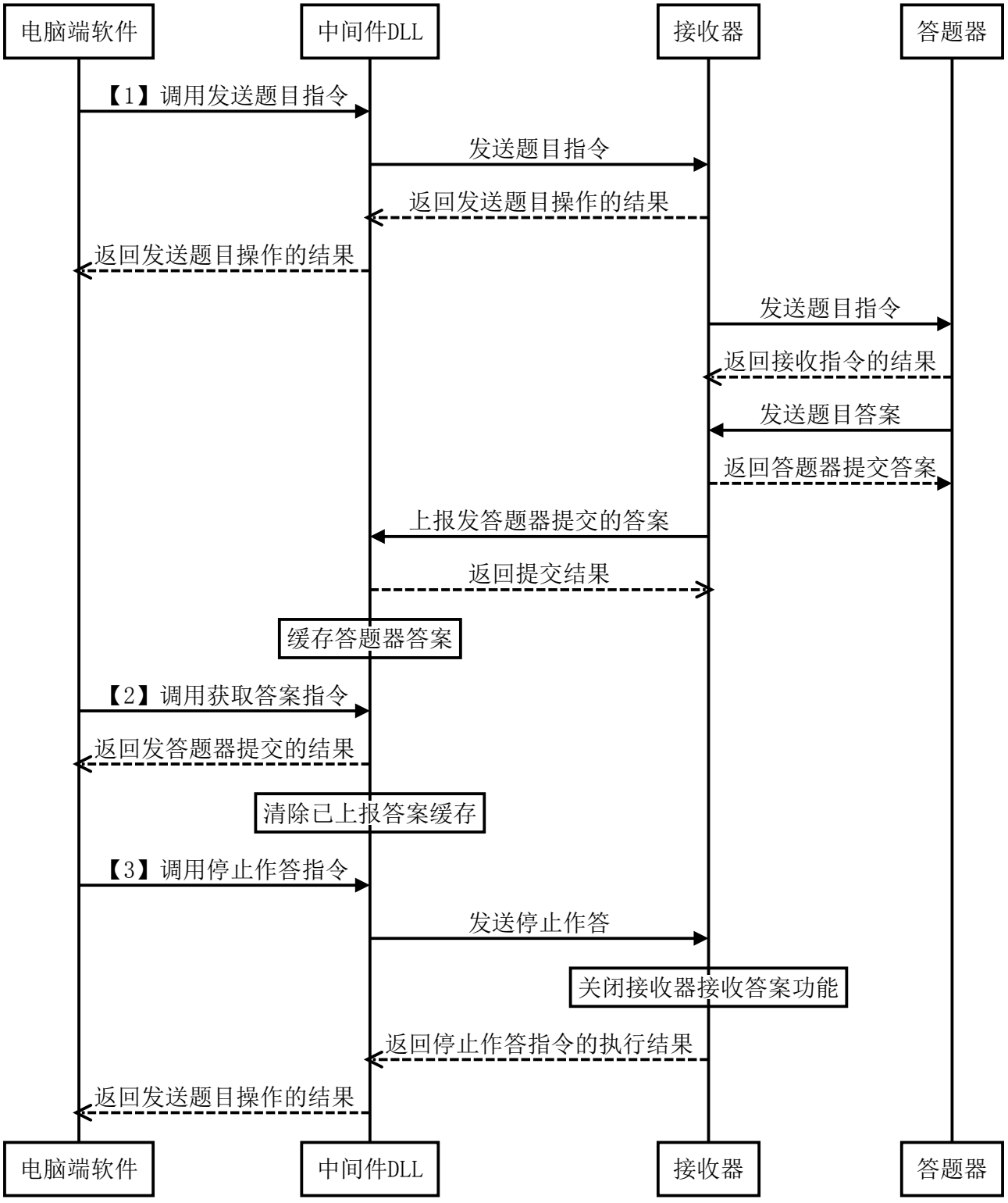
【接收器->DLL指令】：

```
{
    'fun': 'get_device_info',
    'device_id': '3633897184',
    'software_version':  'v0.1.0 ',
    'hardware_version':  'ZL-RP551-MAIN-E',
    'company':  'zkxltech'
}
```

附录1 绑定过程时序图



附录2 答题过程时序图


电脑端软件
中间件DLL
接收器
答题器
【1】调用发送题目指令
发送题目指令
返回发送题目操作的结果
返回发送题目操作的结果
发送题目指令
返回接收指令的结果
发送题目答案
返回答题器提交答案
上报发答题器提交的答案
返回提交结果
缓存答题器答案
【2】调用获取答案指令
返回发答题器提交的结果
清除已上报答案缓存
【3】调用停止作答指令
发送停止作答
关闭接收器接收答案功能
返回停止作答指令的执行结果
返回发送题目操作的结果
电脑端软件
中间件DLL
接收器
答题器