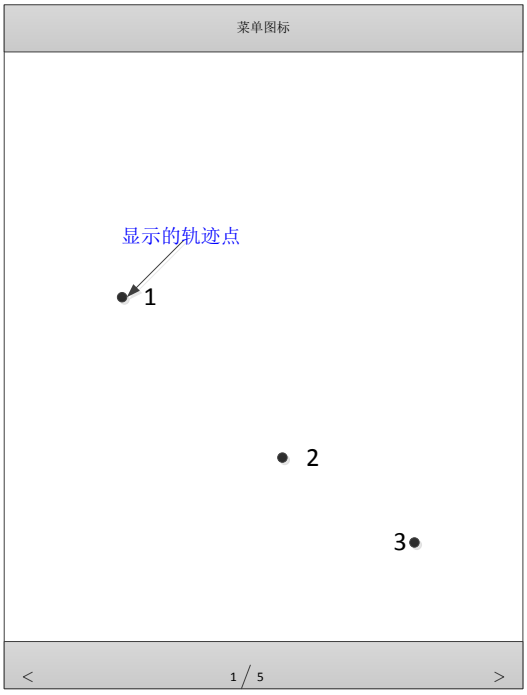
1. 软件需求

根据上述分析，首先，我们在 iOS 系统中实现防掌触处理。

对于书写连线轨迹，显示原始坐标轨迹时折线明显，需要有滤波处理。书写考虑原笔迹要求，软件最终还需要写字的笔峰处理（无压感和带压感模式）算法。

软件 Palm-rejection 处理基本要求包括两个部分：

A）无蓝牙连接时的处理



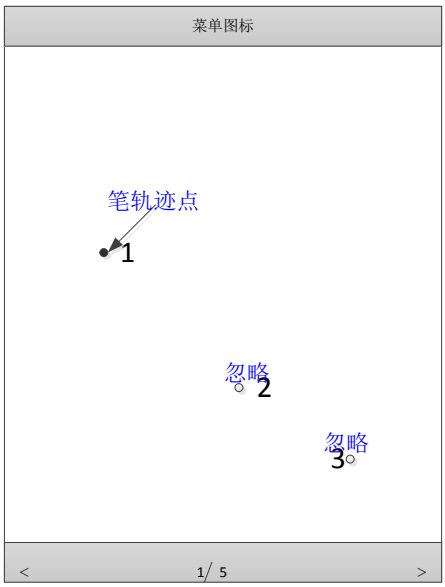
当蓝牙没有连接时：屏幕上只能同时显示一个点的轨迹。

1、当屏幕上只有一个点，屏幕显示这个点的轨迹。

2、当同时屏幕上有两个或者以上的点时，显示的轨迹是最靠近上面的点的轨迹；
同样高度的点显示靠左的点。

3、当屏幕开始一个点显示轨迹时，此时出现其他的点，如果这个点位置比当前点低，忽略此点（直到此点抬起），仍然显示当前点轨迹； 如果此点的位置高于当前点，当前点之前绘制的轨迹（一笔，从落笔到当前）全部删除，同时开始显示新点的轨迹。如果，再来其他的点，依此类推。

B）有蓝牙连接时的处理



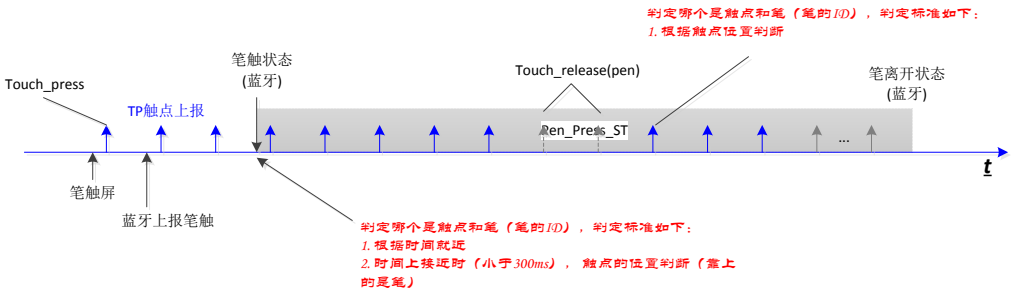
当蓝牙连接时：屏幕上书写区域只能显示笔的轨迹，菜单区域可以接收到笔和手指触摸。

1、没有蓝牙（笔触状态）上报时，笔的识别根据接收蓝牙笔触状态确定：
- 接收到笔触信号时，如果触摸点只有一个，此点确定为笔；(Tp点的开始和笔触时间在1s以内)
- 如果触摸点数量大于1个，时间上和接收到笔触状态近的点是笔（如果时间上相距太近，则以靠上的点作为笔）
-

2、当目前已经笔触状态，笔短时抬起然后放下（蓝牙不会上报笔release），此时以新触点位置判断笔的ID。
- 新触点与release时刻笔的触点位置较近是，新触点判定为笔
- 新触点与release时刻其他触点相距较近， 判定为手

3、考虑蓝牙延时，接收到笔触状态时，此时，笔触已经上报过点了，为了保证笔触是没有开始的几个点丢失，应该预先buffer触点的数据

4、显示轨迹时，会有折线出现。 因此需要增加buffer做轨迹的滤波。



2. 算法结构

主机处理 Palm-rejection 根据屏触点事件和笔通过蓝牙传送的笔触事件为基础实现。之前我们做个算法，碰到的问题是通过 OPS 检测笔触信息不准确，不能很好的反应笔和触屏触点的对应关系。

Adonit 通过压感和电容检测来共同判决的（专利），但是压感是做为一种 option。因此我们调整了笔的算法，是的 OPS 检测更加准确可靠。根据 OPS 信息确定笔触屏（已经测试

了数据判定可靠）。

A） Touch_event 事件处理

根据 Touch 事件处理当前点。

Touch 事件包括 Down、Up、Move 事件。

DOWN：

当 Down 事件：表示一个 Touch 的起点。由于笔检测到触屏信息到蓝牙发出笔触状态到设备收到笔触，存在一定的延时（也就是触屏上已经有笔的触点上报时，经过一定延时才能收到笔触屏的消息），因此需要延时对两个事件作同步处理。

当 DOWN 事件时，

如果 PENID 有效，PENID 的点是笔触的点，其它点忽略。

如果 PENID 无效，PICK-UP 检测。如果不是 PICK_UP，记录当前的时间戳，并将其坐标放入队列中。如果是 PICK-UP，则**当前点的 ID 直接判定为笔（PENID=ID）。**

Pickup 是为了解决笔从触屏状态抬起并且快速放下时的判决。因为此时笔触屏的状态可能不会解除。Pick-up 检测是根据时间和位置检测的。

时间：新触点和笔触 END 点之间的时间间隔在一定间隔内。

位置：新触点和笔触点之间的距离小于一定阈值，且新触点和笔触解除时刻其他触点位置大于一定阈值。

当 UP 事件时，

如果 PENID 有效，解除 PENID，处理笔抬起动作， **记录时间戳和非笔触点的坐标。**

如果 PENID 无效，忽略。

当 MOVE 事件时，

如果 PENID 有效，PENID 的点是笔触的点，其它触点忽略。

如果 PENID 无效，缓存当前触点坐标。

B） Pen_event 事件处理

跟据笔触事件判定当前笔的 ID，即 PENID 赋值。

笔触事件：

笔触 Release 时，清除所有的时间戳和触点 buffer。

笔触 Press 时，判断触点 Buffer 中的点的时间戳和笔触 Press 时间满足一定时间间隔，如果此时有多个点满足，则根据点的位置（靠左）信息确定笔的 ID（**PENID 赋值**）。