APP详细要求

* **UI功能**

界面可以有以下按键：

1. 开始/暂停清扫
2. 开始/结束空净、
3. 停止移动(可手动控制前后左右)
4. 返回基座充电
5. 进入保留区域选择模式

进入保留区域选择后，地图停止旋转，其他按钮消失，仅剩“确定”“取消”按钮，手指在屏幕上划动，则划过的区域被涂抹为灰色，点击“确定”按钮后，将灰色区域的坐标点发送至小车，同时将灰色区域储存，以备下次继续添加区域；取消则不进行任何改动

1. 重设地图旋转方向
2. 定时功能(只摆个样子)
3. 显示运行信息(只摆个样子)
4. 关于(你自由发挥)

点击屏幕，目标红点亮一下后渐变消失，若在消失之前再次点击，则以后一次为准

* **地图管理**

则蓝点闪烁显示当前小车位置

手指拖动可移动地图

两指可缩放地图

地图的旋转主要由陀螺仪决定。有一个按钮是重置旋转，使得角度回到0

注意地图的移动和旋转是记忆的，不受地图更新的影响

由于地图经过了平移、缩放、旋转变换，故手指点击后的坐标也需要进行对应的逆变换，然后才能发送给小车。老版本算错了

* **网路数据传输协议**

请注意，TCP/ IP协议只是对于数据的完整性和顺序做出了保证，但并不对于数据传输的时延和分包进行规定，也就是说，调用一次recv(socket)得到的字符串可能不对应于一次发送的字符串。比如，你发送一个“Move 12345”，可能第一次recv收到的string是“Mov” 第二次是e1234 第三次是“5”和另一次发送的数据。在你以前APP中没有考虑到这种情况，虽然大多数情况下可以运行，但网络信号不好的时候就可能导致通讯协议的混乱，发送大量数据时尤其明显。

具体情况参见http://www.binarytides.com/receive-full-data-with-recv-socket-function-in-c/

为了解决这个问题，我所有的数据回车符“\n”标注结尾。你需要把接收到的数据先放到一个缓冲区内，然后扫描回车符，扫到之后才能认为接收到了一行数据。

网络各种指令、数据均以ASCII字符串格式传输，区分大小写

* APP -> 小车 指令与数据格式:

开始清扫 “Start Clean\n”

暂停清扫 “Pause Clean\n”

停止移动 “Stop Clean\n”

开始空净 “Start Clear\n”

停止空净 “Stop Clear\n”

返回基座充电 “Return\n”

前进 “Forward\n”

后退 “Backword\n”

左转 “Left\n”

右转“Right\n”

目标点“Target %6.3f %6.3f\n”

App退出时自动发送“Exit\n”

传送保留区域点：

先发送“Avoid %d\n”其中%d填入保留点的数量n，然后发送n次“%6.3f %6.3f\n”点的坐标。不必将每个保留点都发过去，只要在区域内每隔几个点发一个关键点即可。

* 小车-> APP指令与数据格式:

小车会每隔几秒向APP发送一次地图(全图) 数据：

第一行“Map %d\n”其中%d是行数n

然后n行“%6.3f %6.3f %6.3f 6.3f %6.3f %6.3f \n” 分别对应五个float变量x1,y1,x2,y2,c

x1 y1 x2 y2即为地图中各条线段的坐标，画出来即可。

变量c目前保留，收到的总是0，以后有用。

若收到“Error %d\n”则表明出错

* **程序框图(粗略)**



* **其他**

APP启动之后每隔4秒尝试一次网络连接，直到连接上或退出为止；

若连接中失去信号，则同上。