结构：

搭车保证车的重心在后面，实在不行在前面加铜柱来保证车在急停不会翻车。

硬件:

1. 手工电路板用于测试，确认电路板后一定要送去工厂做工业板，因为手工板存在的隐患比较多。比如：腐蚀时会出现很多断线的情况，肉眼很难看出来，上电以后发现无法工作，要花费很多的时间去排查。
2. 电路板上功耗大模块的要独立于主控的供电，否则车在运行过程中主控会出现供电不足的情况，导致车跑飞。
3. 电路板上的接口迫不得已不能用排针，排母，尽量用XH这种接头。

程序：

救援车的代码相对来说算挺简单，救援车的控制该份三个部分:

1. 与WIFI模块的串口通信数据的处理。如果有条件一定要买有检验位数据的WIFI模块，这样能轻松的解决丢控后车会疯，如果没有这种模块那就尽量减少手动模式的时间。
2. 自动进房间、自动回起点。在写自动任务路径时，推荐1号房单独一个出发的方向，2、3、4号房同一个出发的方向，这样是最省时间的路径。去2号房的时候有一段依靠墙壁很不好写逻辑的路径最在保证车直行的情况下直接用延时来控制走的距离，从3号房回来也是如此，少用不靠谱的判断条件。
3. 小车从手动模式切换到自动回家的对位、手动模式下撞墙保护。撞墙保护前最好加一段提前减速的代码，否则车会超过你的限位距离导致撞墙，而且不减速一到限位会出现急停的情况，容易前倾翻车。对位时需要写一段控制摄像头的代码，因为在操作手的视角里是没有方向感的全是白色，需要切换不同的视角来判断方向。

灭火车代码：

1. 沿墙校准最好不要用在出口处，因为超声波距离突变大会影响到校准精度，或者行进方向发生突变。
2. 动作的角度一定要标准化！
3. 在转弯之前要调好车身角度，即原地校准函数。
4. 采用节点自动选择函数，按阶段来写，减少代码量

总结：

这次比赛我们尽到了最大的努力去做，没有得特等奖的原因有挺多

第一点：在模块选型的时候没有听学长的建议用了超声波模块来检测距离，因为超声波的刷新频率太慢，导致在写控制代码时出现了很多问题，下一届的传感器一定要选刷新频率高一点的模块。如果明年还用超声波方案的话，建议使用串口超声波，数据稳定，频率高，并且一个串口串联多个超声波！

第二点：选择两个队伍合作的方式进行比赛，这是很不成熟的左比赛方式，最后的竞争还是我们队内的竞争。