model.c是所有模块共有的函数文件，包括每个模块的main函数也在这个文件之中。modle.c总共包含数十种函数，具体的记不清楚，也没有代码可以查看，就简单的介绍一下大概的功能函数：

1、与驱动driver通信的功能，接收输入输出端口信息，间接接收来自用户的开始与退出等指令；

2、与指挥家进行通信，接收来自其他模块的临时命令；

3、与监控通信，包括心跳包以及各种监控信息的发送；

4、模块初始化，接收来自connect表的基本参数，比如ip和端口等；

5、数据接收函数；

6、数据发送函数；

7、程序运行main函数，每个模块的运行流程一致，只需要重构func功能函数即可；

8、磁盘检测函数，每个模块在写入文件之前，可以调用此函数检测磁盘空间是否安全；

9、信号处理函数，每个模块可以自行监测信号，并做出处理；

10、重构printf函数，主要是方便python界面的显示，所以对printf进行了重构。

model.c文件是整个框架最直观的体现，也是除了算法以及arm之外代码量最大的文件。由于每个模块的基础功能完全一致，每个功能的执行流程也完全一样，所以可以把它们封装为同一个文件，类似于一个库，实际上也可以把这个文件编译为一个动态库。model.c为每一个模块都注册了一个func函数，这也是每个模块的核心功能，在模块开发时基本上只需要重构这个核心功能即可，其它的诸如申请端口，连接输入输出，心跳发送等都在model.c里面进行完成。除此之外，model.c还为所有模块提供了很多方便的接口，主要是接收以及发送各种数据或者信息。

model.c无论在哪种软件框架之中，都是存在的且必须的，因为一套软件之中必定会有大量的重复行为，这个时候从代码上一定要把它们统一起来。尤其是我们这种模块式软件，很多功能被切割为多个进程，多个进程之间的统一联系就是modle.c，由于整个软件被分为多个进程（模块），所以model.c里面包含了很多重要的功能，这些功能一旦出现问题，会导致所有模块不能正常工作。model.c的这些功能函数对于资源的把控也非常重要，因为这些功能的所消耗的资源会因位模块数量的增多而不断增加，在model.c里面增加一个线程，相当于在整个软件之中增加了几十个线程，在model.c里面申请1M的空间，实际上整个软件会申请几十M空间……