robot.c中，首先运行的是init\_robot()函数，将控制器相关的一些变量初始化，之后的do\_power\_monitor()函数实现了电池电压的监控，do\_timer主要是利用中断周期，来对pid进行运算，来改变一些接下来的函数中需要用到的变量。do\_run()函数先定义了mode\_count和start\_beep\_flag，mode\_count可以实现通过观察色彩灯是否熄灭来判断状态之前是否进入了蓝牙模式，start\_beep\_flag可以实现通过灯光闪烁的次数来显示机器人所处的状态；对一些timer进行初始化之后，进入到了while循环。

进入while循环之后，首先判断电池电压是否是满足条件，电池电压降低到设定阈值1 则蜂鸣器报警 降到阈值2则停止机器人。停止机器人的代码可以用于其他的需要停止机器人的操作中，可以直接套用。

检查完机器人的电池电压大小之后，设置了一个心跳周期。

通过一个switch操作，对我们输入的g\_robot.mode进行分情况操作：

如果是BLUETOOTH\_MODE状态，则改变mode\_count的值，打开LED灯并显示为蓝色；

如果是NORMAL\_MODE状态，则进行例行的mode\_count的判断，接受数据以及通讯超时中断操作，以及发送数据包之后将模式修改为接受模式，但是对该模式的实际作用不是很理解；

如果是JOYSTICK\_MODE状态，进入手柄控制模式，则先将LED灯显示为绿色，再进行对手柄输入数据的解压，以及通讯超时中断操作；

如果是SELFTEST\_MODE状态，进入自检模式，则先将LED灯显示为白色，再进行一系列的操作验证小车功能；

如果是DEBUG\_MODE状态，进入PC机下发参数模式，则先将LED灯显示为黄色，再根据USB\_Receive\_Buffer[0]的值进行分情况操作，主要是将自身的状态数据传输给PC机。

PID\_TUNE\_MODE、SHOOT\_POWER\_CURVE\_TUNE\_MODE、NORMAL\_MODE代码基本一致，不太了解这样设置的意义；

如果是ACTION\_TUNE\_MODE状态，我的理解是测试函数，具体的功能并不太清楚。

switch操作结束后，一个while循环基本结束。

最后设置了一个SysTick\_Handler中断程序，并利用看门狗来使系统定期重置。

以上就是我对robot.c文件的理解。