**紧急刹车模块使用说明**

**简述：**

主要独立于车控系统，如果车控异常，紧急刹车模块会发出急停命令，让所有的车都急停下来，避免撞车风险。

紧急刹车模块的框图如下：

KEY0退出急刹车模式

定时接收心跳包

发送急刹车命令

统计/打印

FS

控

制

yes

no

Can通信

Can通信

TM1定时器

KEY1进入急刹车模式

在正常收到五个心跳包后，才进入急停；

如果进入急停后，在正常收到五个心跳，

才开始记录

刹车控制器

主要功能：就是判断是否在定时周期里面接收到来自车控的心跳，如果没有收到则响应紧急刹车模式。

**紧急刹车的逻辑如下：**

定时接收心跳包

统计/打印

yes

no

TM1定时器

正常收到五包心跳？

发送急刹车命令

退出急刹车模式

正常收到五包心跳？

yes

只判断第一次

在正常收到五个心跳包后，才进入急停；如果进入急停后，在正常收到五个心跳，才开始记录。

**接口使用说明：**

获取数据

|  |  |
| --- | --- |
| 接口名称 | 向车控系统周期请求CAN数据的接口  u8 Can\_Receive\_Msg(u8 \*buf)； |
| 接口描述 | 主要在定时中断里面，周期时间内接收来自车控系统发送CAN心跳包； |
| 请求参数 | buf：用于缓存车控发送过来的数据 |
| 备注：车控传过来的数据内容不限，但是数据长度必须为8！ |
| 返回参数 | 返回值为0：无数据被收到 |
| 返回值为其它：表示接收到数据的长度 |

推送数据

|  |  |
| --- | --- |
| 接口名称 | 向车控系统周期推送CAN数据的接口  u8 Can\_Send\_Msg(u8\* msg,u8 len)； |
| 接口描述 | 主要在定时中断里面，周期时间内没有接收来自车控系统的CAN心跳包，则触发急停模式；向车控发送急停命令。 |
| 推送参数 | msg：用于缓存向车控发送的数据  msg[5] = 255 & 0xff;  msg[6] = (255>>8) & 0xff;  msg[7] = 2; |
| len:数据长度，最大为8 |
| 返回参数 | 返回值为0：发送成功 |
| 返回值为其它：发送失败 |

**函数使用说明：**

心跳接收

|  |  |
| --- | --- |
| 函数名称 | void recv\_heart(u8 \*buf); |
| 函数描述 | 定时器1，收到心跳Can\_Receive\_Msg(canbuf)（从CAN收到ID为1的包）；正常收到心跳包，开始记录收到的时间以及次数，然后都写入flash，日志系统还在开发。 |
| 函数参数 | buf：从车控传过来的数据包。 |
| len:数据长度，最大为8。 |
| 返回参数 | 无 |
|  |

紧急刹车

|  |  |
| --- | --- |
| 函数名称 | void urgency\_stop(void); |
| 函数描述 | 如果定时器1在两秒内，没有收到心跳包；将会通过CAN发送msg。  msg[5] = 255 & 0xff;  msg[6] = (255>>8) & 0xff;  msg[7] = 2;  Can\_Send\_Msg(msg,8); |
| 函数参数 | 无 |
|  |
| 返回参数 | 无 |
|  |

刹车模块关闭重启

|  |  |
| --- | --- |
| 函数名称 | void key\_brake(void); |
| 函数描述 | KEY0退出急刹车模式  KEY1进入急刹车模式 |
| 函数参数 | 无 |
|  |
| 返回参数 | 无 |
|  |