

|  |  |
| --- | --- |
| **保密级别** | **秘密级** |
| **版本号** | **V0.1** |
| **归档部门** |  |

**设计文档**

**【车联网】**

**硬件研发部**

**二〇一九年一月**

# 模块框架

模块采用底层驱动和上层应用的方式实现软件编写。

需实现内容：

* 获取车速
* 获取里程数
* 获取位置信息
* 获取故障信息
* 信号灯指示
* 离线存储
* 参数配置

### 底层驱动

底层驱动分为：

* Mc20驱动
* CAN驱动
* LED驱动
* 系统驱动

### 上层应用

* 配置参数
* 上报数据
* 离线记录
* 状态检测

# 底层实现

### 2.1 MC20驱动

#### 2.1.1 包含函数

1. mc20硬件初始化
2. mc20指令集
3. mc20 GPRS开启
4. mc20 GPS开启
5. mc20消息队列
6. mc20接收数据

#### 2.1.2 初始化内容

1. Void Mc20\_Init(void);

#### 2.1.3 运行事件

1. Void MC20\_Core\_Run\_Process(void);

此函数20ms调用一次

1. Void MC20\_Core\_Rev\_Loop\_Process(void)

此函数为触发事件

1. Void MC20\_GPRS\_Start\_Process(void);

此函数为触发事件

1. Void MC20\_GPS\_Start\_Process(void);

此函数为触发事件

1. Void MC20\_GPS\_GetLocation\_Process(void);

此函数为触发事件

#### 2.1.4 结构体

Typedef struct

{

Uint8\_t status\_machine;

Uint8\_t status\_Rev;

}Status\_Mactine\_t;

Typedef stuct

{

Status\_Mactine\_t GPRS\_Status\_Machine;

Status\_Mactine\_t GPS\_Status\_Machine;

}MC20\_CoreStatus\_t;

#### 2.1.5 接口

Void MC20\_GPRS\_Start(void);

Void MC20\_GPS\_Start(void);

Void MC20\_GPS\_GetLocation(uint16 longitude,uint16\_t latitude);

### 2.2 CAN驱动

#### 2.2.1 包含函数

* CAN初始化
* CAN获取车辆信息
* CAN消息队列

#### 2.2.2 初始化内容

Void CAN\_Init(void)

#### 2.2.3 运行事件

1. Void CAN\_Data\_Loop\_Run(void);

此函数1s调用一次

1. Void CAN\_Data\_Analysis\_Porcess(void);

此函数为触发事件

#### 2.2.4 结构体

Typedef struct

{

Uint16\_t Value;

Uint8\_t status;

}CAN\_Car\_Info\_t;

Typedef struct

{

CAN\_Car\_Info\_t mileage;

CAN\_Car\_Info\_t Speed;

}CAN\_Kind\_info\_t;

#### 2.1.5 接口

Void CAN\_Start\_Get\_Mileage(void);

Void CAN\_Start\_Get\_Speed(void);

Uint16\_t CAN\_Get\_Mileage(void);

Uint16\_t CAN\_Get\_Speed(void);.

### 2.3 LED驱动

#### 2.3.1 包含函数

* LED初始化
* LED频率设置
* LED运行

#### 2.3.2 初始化内容

Void LED\_Init(void)

#### 2.3.3 运行事件

Void

#### 2.3.4 结构体

#### 2.3.5 接口

### 2.4 系统驱动

#### 2.4.1 包含函数

参数保存修改

离线记录

#### 2.4.2 初始化内容

Void System\_Paramter\_Load(void);

#### 2.4.3 运行事件

#### 2.4.4 结构体

#### 2.4.5 接口

Void Offline\_Record\_Info(uint8\_t \* buf,uint16\_t len);

Void Offline\_Read\_Info(uint8\_t num,uint8\_t \*buf,uint16\_t len);

# 上层实现

### 3.1 初始化内容

Void User\_Open\_Func(void)

### 3.2 运行事件

Void User\_Scan\_Status\_Loop(void);