**软件概要设计说明书**

**2015年 02月 05日**

|  |
| --- |
| 1 引言  1.1 系统简介  使用前后台系统的状态机实现当前系统。  1.2 设计目标    2 总体设计  2.1 MCU硬件结构    2.2 软件系统设计架构  软件系统设计架构分为四个层次：   1. 驱动层； 2. 外设模块驱动层； 3. 模块应用层； 4. 应用层。     2.3 模块流程  2.3.1 上电初始化      3 接口设计  3.1 串口驱动接口  vUart0Init();  BYTE byUart0Read();  void vUartWrite(BYTE data);  3.2 ApuDriver接口  void vApuDrvInit();  void vApuDrvTask();  void vApuDrvNRead(BYTE\* pData);  void vApuDrvNWrite(BYTE\* pData, BYTE length);  3.3 ApuMessage接口  void vApuMessageInit();  void vApuMessageTask();  void vApuMessageNWrite(BYTE\* pMsg, BYTE length);  3.4 IIC驱动接口  void vIICInit();  BYTE byIICRead();  void vIICWrite(BYTE data);  3.5 eeprom驱动模块接口  void vEepromInit();  void vEepromWrite(int iAddr, BYTE data);  BYTE byEepromRead(int iAddr);  void vEepromNWrite(int iAddr, BYTE\* pData, BYTE length);  void vEepromNRead(int iAddr, BYTE\* pData, BYTE length);  3.6 TEF6638驱动模块接口  void vTef6638Init();  //待补充  3.7 收音机模块接口  收音机对外接口显示信息的输出以及控制信息的输入:  void vRadioTask();  void vRadioInit();  void vRadioTimer();  void vRadioApuHandle(BYTE\* pMsg, BYTE length);  void vRadioKeyHandle(int iKey);  3.8 音频模块接口  void vAudioTask();  void vAudioInit();  void vAudioTimer();  void vAudioApuHandle(BYTE\* pMsg, BYTE length);  void vAudioKeyHandle(int iKey);  3.9 I/O驱动(io.h)  3.10 AD驱动接口  void vADInit();  BYTE byADRead();  3.11 Key接口  void vKeyInit();  void vKeyTask();  void vKeyTimer();  3.12 CanDriver驱动接口  void vCanDrvTask();  void vCanDrvInit();  void vCanDrvTimer();  发送消息结构:  Typedef struct  {  CanIdType id;  BYTE length;  PreWritePtr pxPreWriteData;  ConfirmationPtr pxConfirm;  BYTE\* pData;  BYTE offset;  }CanWriteMsg;  接收消息结构:  Typedef struct  {  xCanIdType id;  BYTE length;  PreCopyPtr pxPreCopy;  IndicationPtr pxIndication;  BYTE\* pData;  BYTE offset;  }CanReadMsg;  3.13 Timer驱动接口  void vTimerInit(BYTE TickUnitMS);  BOOL bTimerTick(void);  3.14 SoftTimer驱动接口  void vSoftTimer(void);  4 环境配置  4.1 类型定义(typedef.h)  typedef unsigned char BYTE,UINT08,BOOL; //无符号8位整型变量  typedef signed char INT08; //有符号8位整型变量  typedef unsigned short WORD,UINT16; //无符号16位整型变量  typedef signed short INT16; //有符号16位整型变量  typedef unsigned long DWORD,UINT32; //无符号32位整型变量  typedef signed long INT32; //有符号32位整型变量  typedef unsigned int UINT; //无符号整型变量  typedef signed int INT; //有符号整型变量  5 系统出错处理设计  5.1 出错信息  用图表的方式说明每种可能的出错或故障情况出现时，系统输出信息的形式、含意及处理方法。  5.2 补救措施  说明故障出现后可能采取的变通措施。  5.3 系统维护设计  为了系统维护的方便而在程序中专门安排用于系统的检查与维护的一些检测点和专用模块。 |
|  |