1. 软件结构

Main

Timer

power

transfer

storage

memory

SPI

UART

LED

ADC

功能模块

驱动模块

驱动模块：直接操作寄存器，完成简单的单一功能。

功能模块：调用驱动模块提供的接口函数，实现程序的逻辑功能。

1. 模块设计
2. 主程序

完成系统初始化和各模块初始化。在每100ms的时钟中断的间隔时间里，完成电源监控、数据传输和存储。

1. main

1）原型：void main(void)

2）输入：无

3）输出：无

4）返回值：无

5）返回值：无

5）描述：

1. 电源

实现对电源监测，生成报警信息。实现对输入电源的判断，点亮相应的LED指示灯。实现内部电池充电管理。

1. pwr\_initialise

1）原型：void initialize(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：电源模块初始化。

1. pwr\_monitor

1）原型：void pwr\_monitor(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：通过对各个电源监测点的电压AD值的分析，判断是否存在报警。同时也进行当前供电电源的判断和内部电池的充电管理。

1. pwr\_detect\_source

1）原型：void pwr\_dectect\_source(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：通过分析供电通路的压降情况，判断当前由那一个电源供电，点亮相应的LED指示灯，更新当前供电情况信息。

1. pwr\_charge\_monitor

1）原型：void pwr\_charge\_monitor(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：计算充电电流，分析电池电压和充电电流，判断充电情况，控制是否充电，更新充电信息。

1. pwr\_translate

1）原型：static char pwr\_translate(void)

2）输入：无

3）输出：无

4）返回值：0x0，平均值计算过程中，数据不可用；

0x1，数据可用。

5）描述：模块内部函数，将AD值转换成真实电压值，并做均值去抖处理。

1. pwr\_charge\_enable

1）原型：pwr\_charge\_enable

2）输入：无

3）输出：无

4）返回值：无

5）描述：宏定义，可当作内部函数使用。使能电池充电。

1. pwr\_charge\_disable

1）原型：pwr\_charge\_disable

2）输入：无

3）输出：无

4）返回值：无

5）描述：宏定义，可当作内部函数使用。停止电池充电。

1. 传输
2. 存储
3. 定时器

提供等间隔中断，帮助系统做精确时间控制。

1. timer\_initialise

1）原型：void timer\_initialise(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：初始化定时器A0和A1。其中A0产生10Hz的中断给系统使用，A1产生2Hz的中断用于控制LED灯闪烁。

1. timer\_a0\_start

1）原型：void timer\_a0\_start(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：定时器A0开始。

1. timer\_a0\_stop

1）原型：void timer\_a0\_stop(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：定时器A0停止。

1. timer\_a1\_start

1）原型：void timer\_a1\_start(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：定时器A1开始。

1. timer\_a1\_stop

1）原型：void timer\_a1\_stop(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：定时器A1停止。

1. timer\_a0\_isr

1）原型：\_\_interrupt void timer\_a0\_isr(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：定时器A0中断服务函数。对所有AD通道采样，更新AD采样完成标志。

1. timer\_a1\_isr

1）原型：\_\_interrupt void timer\_a1\_isr(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：定时器A1中断服务函数。根据需要改变红色报警灯和黄色报警灯状态，实现灯的闪烁。

1. ADC
2. adc\_initialise

1）原型：void adc\_initialise(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：ADC模块初始化。

1. adc\_start

1）原型：unsigned int adc\_start(unsigned int adc\_signal, unsigned int \*result, unsigned int timeout)

2）输入：adc\_signal，ADC通道号；

timeout，ADC采样等待超时时间。

3）输出：result，存储ADC转换值的变量指针。

4）返回值：ADC\_RESULT\_OK，函数执行成功；

ADC\_RESULT\_TIMEOUT，等待超时。

5）描述：对指定ADC通道进行采样，将转换值存放到指定变量。

1. LED

控制系统LED灯。

1. led\_initialise

1）原型：void led\_initialise(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：LED模块初始化。

1. led\_acled\_on

1）原型：void led\_acled\_on(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：点亮直流电源指示灯。

1. led\_acled\_off

1）原型：void led\_acled\_off(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：熄灭直流电源指示灯。

1. led\_dcled\_on

1）原型：void led\_dcled\_on(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：点亮备份电池指示灯。

1. led\_dcled\_off

1）原型：void led\_dcled\_off(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：熄灭备份电池指示灯。

1. led\_btled\_on

1）原型：void led\_btled\_on(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：点亮内部电池指示灯。

1. led\_btled\_off

1）原型：void led\_btled\_off(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：熄灭内部电池指示灯。

1. led\_aled\_r\_on

1）原型：void led\_aled\_r\_on(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：点亮红色报警灯。

1. led\_aled\_r\_off

1）原型：void led\_aled\_r\_off(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：熄灭红色报警灯。

1. led\_toggle\_aled\_r

1）原型：void led\_toggle\_aled\_r(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：切换红色报警灯状态。

1. led\_flash\_aled\_r

1）原型：void led\_flash\_aled\_r(unsigned int switcher)

2）输入：switcher，闪烁开关。

3）输出：无

4）返回值：无

5）描述：开始或结束红色报警灯闪烁。

1. led\_aled\_y\_on

1）原型：void led\_aled\_y\_on(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：点亮黄色报警灯。

1. led\_aled\_y\_off

1）原型：void led\_aled\_y\_off(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：熄灭黄色报警灯。

1. led\_toggle\_aled\_y

1）原型：void led\_toggle\_aled\_y(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：切换黄色报警灯状态。

1. led\_flash\_aled\_y

1）原型：void led\_flash\_aled\_y(unsigned int switcher)

2）输入：switcher，闪烁开关。

3）输出：无

4）返回值：无

5）描述：开始或结束黄色报警灯闪烁。

1. 内存管理

在FRAM中开辟一段空间用于存储报警信息（时间，报警源，电压值）。实现内存的读、写、擦除功能。

1. mem\_initialise

1）原型：void mem\_initialise(void)

2）输入：无

3）输出：无

4）返回值：无

5）描述：内存管理模块初始化。

1. mem\_read

1）原型：int mem\_read(char \* mem\_addr,int size,char \* rdata)

2）输入：mem\_addr，读取的内存起始地址；

size，读取的数据大小。

3）输出：rdata，存储读取数据的变量地址。

4）返回值：MEM\_PARAM\_INVALID，输入变量错误；

正整数，实际读到的数据大小。

5）描述：从FRAM中读取数据。

1. mem\_write

1）原型：int mem\_write(char \* mem\_addr,int size,char \* wdata)

2）输入：mem\_addr，写入的内存起始地址；

size，写入的数据大小；

wdata，待写入的数据地址。

3）输出：无

4）返回值：MEM\_PARAM\_INVALID，输入变量错误；

正整数，实际写入的数据大小。

5）描述：向FRAM中写入数据。

1. mem\_erase

1）原型：int mem\_erase(char \* mem\_addr,int size)

2）输入：mem\_addr，擦除的内存起始地址；

size，擦除大小；

3）输出：无

4）返回值：MEM\_PARAM\_INVALID，输入变量错误；

正整数，实际擦除大小。

5）描述：擦除FRAM中的数据。

1. mem\_clean

1）原型：int mem\_clean(void)

2）输入：无

3）输出：无

4）返回值：MEM\_PARAM\_INVALID，内部调用函数错误；

正整数，可用内存空间大小。

5）描述：擦除整个可用内存。

1. SPI接口
2. 串行口