

# 模拟交通灯控制板用户手册



Technology for easy living

凌阳科技股份有限公司  
SUNPLUS TECHNOLOGY CO., LTD.



---

---

## 目 录

---

---

目 录	3
前 言	4
基本参数	5
模拟交通灯控制板介绍	6
实验举例	8
模拟交通灯控制板原理图	17

---

## 前 言

---

模拟交通灯控制就是使用单片机、PLC 或其它器件来控制一些 LED 和数码管，模拟真实交通灯的功能。红、黄、绿交替闪亮，倒计时显示时间，方向灯指示方向等。

很多高校都有模拟交通灯实验，但没有专门的模拟交通灯控制板。这就给实验室增加了不少麻烦。为了解决这个问题，我们推出的这个模拟交通灯控制板。它可以让学生轻松完成模拟交通灯的基本实验后，有条件模拟更为复杂的情况，完成与实际路口相差无几的交通控制。

---

---

## 基本参数

---




---

产品型号：TRAFFIC BOARD V1.0

工作电压：5.0V

外形尺寸：140mm × 160mm

## 模拟交通灯控制板介绍

下图是模拟交通灯控制板布局示意图。图中的  表示 2 位 7 段的 LED 数码管（用作倒计时显示），<sup>B</sup> 表示双色 LED（用作红黄绿灯）， 表示小按键（用来模拟车流）。

这是一个典型的十字路口，分别用 1、2、3、4 表明四个流向的主车道，用 L、S、R、P 分别表示各主车道的左行车道、直行车道、右行车道以及人行横道。

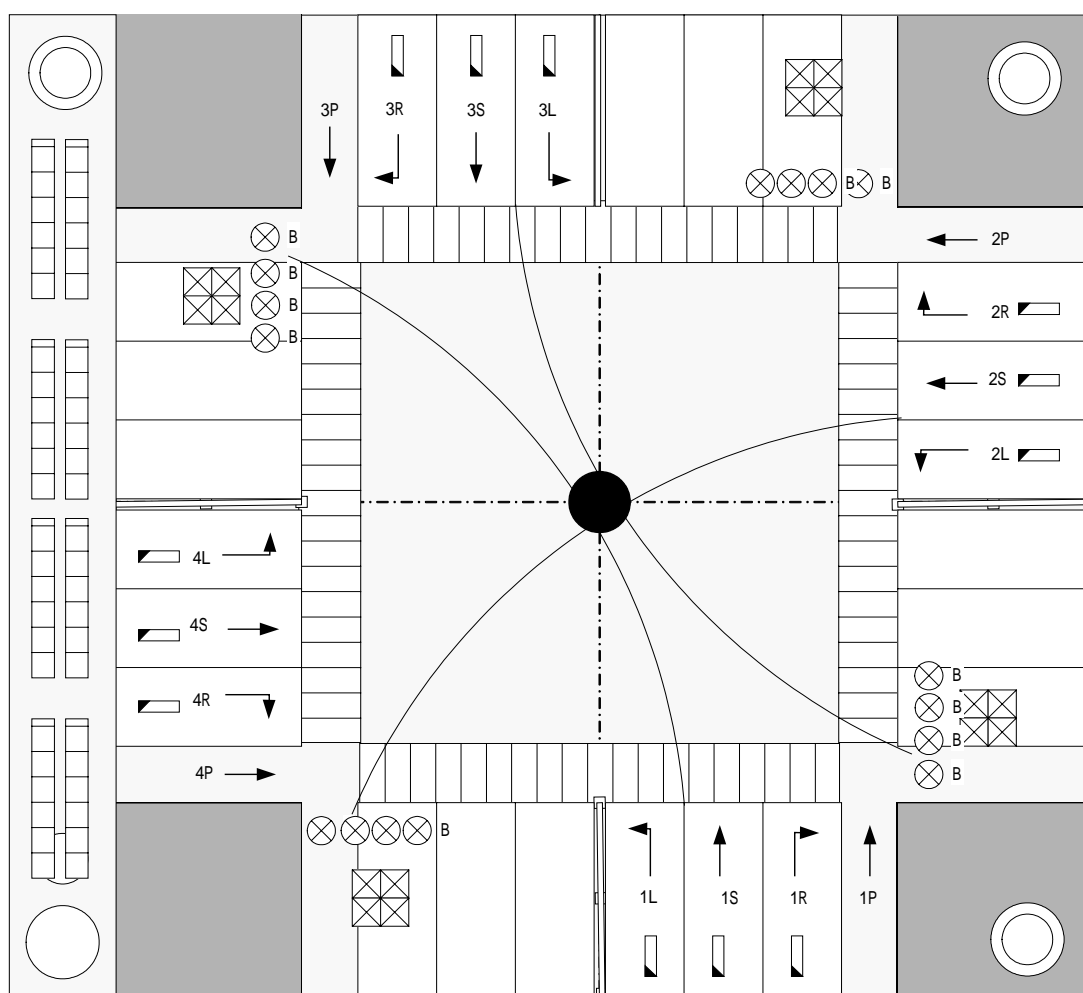


图1 模拟交通灯控制板布局示意图

通过分析很容易得知，除了四个右行车道外，在同一时间，最多只能有两个车道通行，如 1L、

1S 通行时,其它车道都会被阻断。所以在设计红绿灯时,可以两两组合,共有四组(如 1L-1S、2L-2S、3L-3S、4L-4S);而各车道的红灯时间和人行横道通停时间都由这四个组合的绿灯时间决定。通行顺序如下图所示。

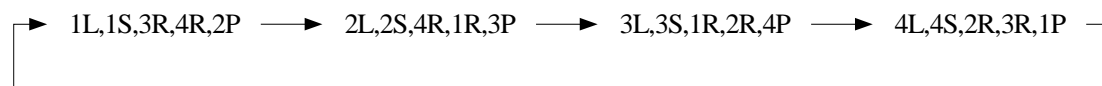


图2 通行顺序

表1 交通灯控制板引脚说明

引脚符号	类型	说明
VCC	P	5V
VSS	P	GND
CS5	O	选通 LED 数码管的高位
CS4	O	选通 LED 数码管的低位
CS3	O	选通第四组双色 LED
CS2	O	选通第三组双色 LED
CS1	O	选通第二组双色 LED
CS0	O	选通第一组双色 LED
IO-0, IO-1, IO-2, ..., IO-7	O	数码管对应的段, 和双色 LED 共用
S-L、S-S、S-R、E-L、E-S、E-R、 N-L、N-S、N-R、W-L、W-S、 W-R、	I	路口的按键输入, -前的 S、N、E、W 分别表示南、北、东、西方向, -后的 L、S、R 分别表示左转、直行、右转

## 实验举例

本系统由一块 61 板和一块自制的模拟交通灯控制板组成，可以模拟比较复杂的路口的交通灯控制。61 板和模拟交通灯控制板通过排线连接。这里先简单介绍系统的结构，具体电路请参考后面的电路原理图。

系统结构图如下：

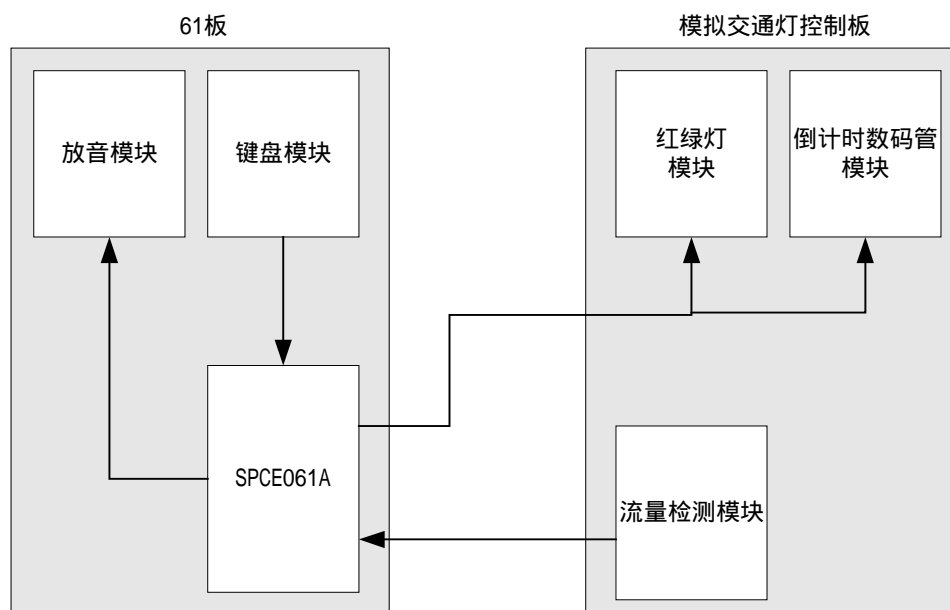


图3 模拟交通灯控制系统结构图

61 板与模拟交通灯控制板的接线方法如下：

61 板	交通灯板
+5V	VCC
GND	VSS

IOA4	W_L
IOA5	W_S
IOA6	W_R
IOA7	E_L
IOA8	E_S
IOA9	E_R
IOA10	N_L



IOA11      N\_S  
IOA12      N\_R  
IOA13      S\_L  
IOA14      S\_S  
IOA15      S\_R

IOB0        CS0  
IOB1        CS1  
IOB2        CS2  
IOB3        CS3  
IOB4        CS4  
IOB5        CS5

IOB8        IO-0  
IOB9        IO-1  
IOB10       IO-2  
IOB11       IO-3  
IOB12       IO-4  
IOB13       IO-5  
IOB14       IO-6  
IOB15       IO-7

系统以一片 SPCE061A 为核心，检测键盘模块和流量检测模块，根据检测结果去按照程序设定的方式去控制红绿灯模块、倒计时数码管模块和放音模块。

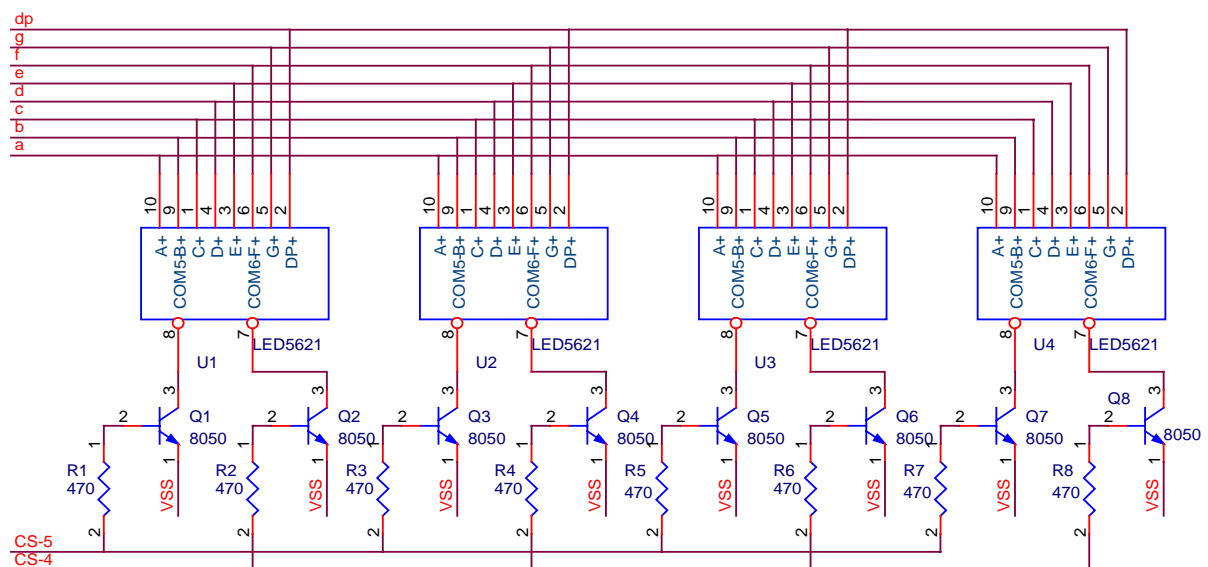


图4 倒计时 LED 数码管电路

数码管完成倒计时显示功能。拿南北方向举例，数码管从绿灯的设置时间最大值往下显示，每秒钟减一，一直减到 1。然后又从红灯的设置时间最大值往下显示，每秒钟减一，一直减到 1。接下来又是显示绿灯时间。如此循环。

系统共有 4 个两位的 LED 数码管，分别放置在模拟交通灯控制板上的四个路口。因为四个方向的数码管应该显示同样的内容，所以我们可以把它们同样对待。也就是说各个方向的数码管个位（把数码管第二位定义为个位，第一位定义为十位）用一根信号线去控制，十位用另一根信号线去控制。这里采用动态显示，段选信号线为 a-dp，位选信号为 CS-4 和 CS-5。

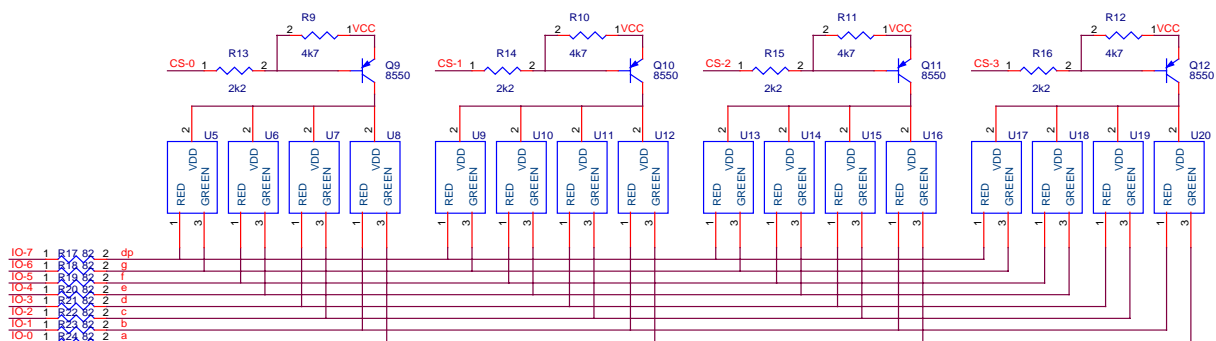


图5 红绿灯双色 LED 电路

双色 LED 可以显示红色、绿色和黄色，可以用作红绿黄灯。我们可以把 16 个 LED 分成 4 个组，东西南北每个方向的灯为一组。每组 LED 的数据线和倒计时数码管的段选线共用，通过 CS-0 到 CS-3 去选通。每个方向 4 个灯，分别是左转弯灯、直行灯、右转弯灯和人行道灯。正常运行时，这些双色 LED 的动作过程和实际路口一致。系统有一种特殊情况，那就是当紧急情况发生时，四面都是红灯，只允许急救车通过。需要说明的是数码管是共阴的，而这里的 LED 是共阳的，编程时需要注意。

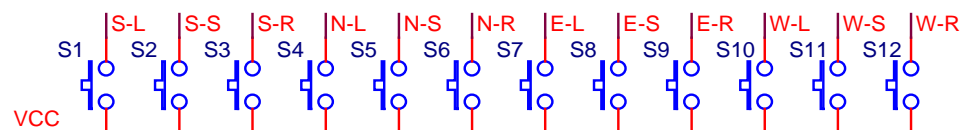


图6 模拟车流检测电路

四个路口的各个车道（不包括人行道）均有一个小按键，用来模拟有车通过的情况。这 12 个小按键在交通灯控制板上表现为两两的焊点，用导体接触才可以导通。每导通一次被认为是有一辆车通过。

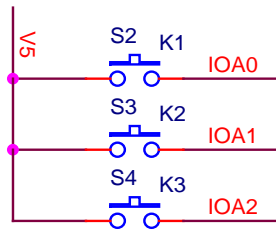


图7 键盘电路

除了模拟车流量的焊点（当作按键来用），模拟交通灯控制板上没有按键，所有设置均通过 61 板的 3 个按键完成。

3 个按键在系统的 4 种状态下有不同的含义。在正常运行状态，按下 Key1 会进入紧急状态，按下 Key2 会进入设置状态，按下 Key3 会进入查询状态。在设置状态，按下 Key1 开始设置南北的红灯时间，按下 Key2 设置南北的绿灯时间，按下 Key3 返回正常运行状态。在查询状态，按下 Key1 或者 Key2 可以让系统播报各个路口的车流量，按下 Key3 返回正常运行状态。在紧急状态，按下 Key3 键返回正常运行状态。

交通灯换向时有语音提示。

放音利用的是 SPCE061A 内部的 DAC。图中的 SPY0030 是凌阳公司的产品。和 LM386 相比，SPY0030 还是比较有优势的，比如 LM386 工作电压需在 4V 以上，SPY0030 仅需 2.4V 即可工作（两颗电池即可工作）；LM386 输出功率 100mW 以下，SPY0030 约 700mW。其他特性请参考其 data sheet。

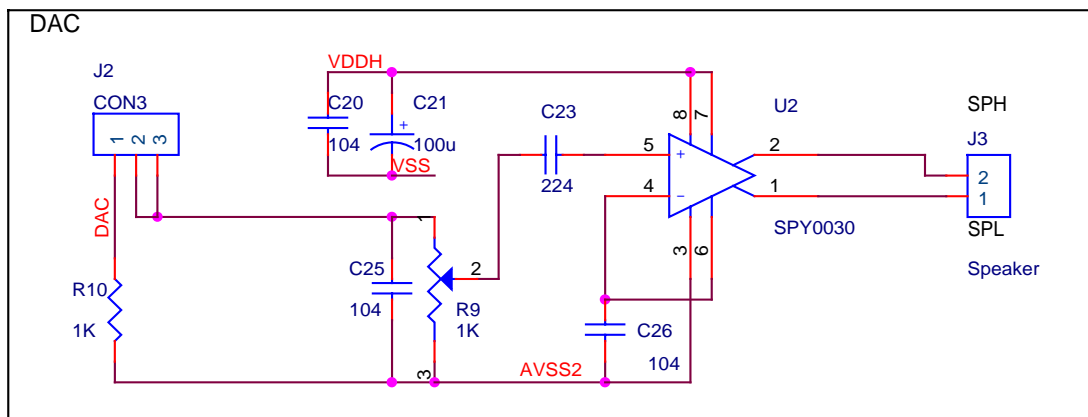


图8 放音电路

本软件系统设计简单。下面给出了相应的软件流程图。

需要源代码的请到凌阳科技大学计划网站（<http://www.unsp.com.cn>）下载。

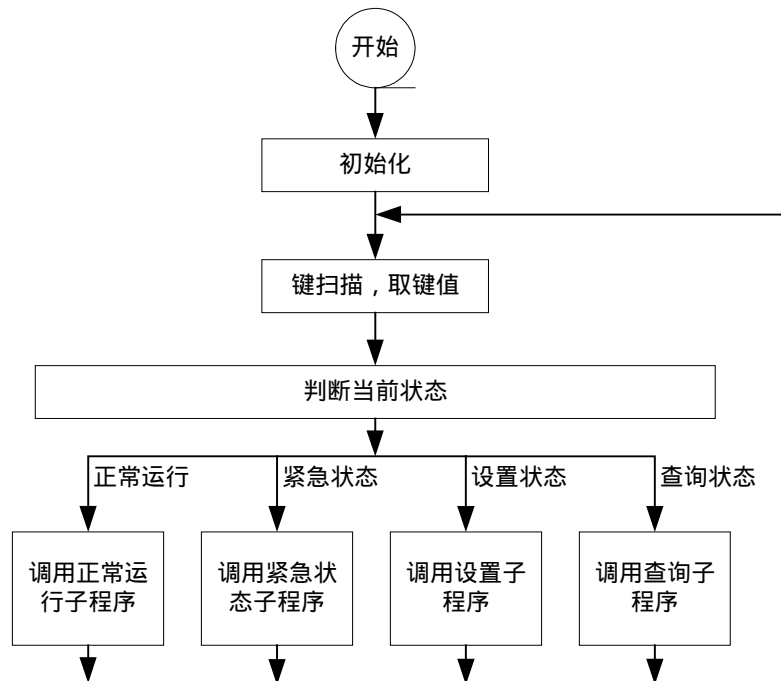


图9 主程序流程图

主程序比较简单，初始化完成后，调用按键扫描程序，取得键值，并根据当前系统状态调用相应的子程序。这里有四个基本的子程序供调用，分别对应系统的失重状态。这四个子程序是正常运行子程序、紧急状态子程序、设置子程序和查询子程序，它们的流程图如下面的几个图所示。

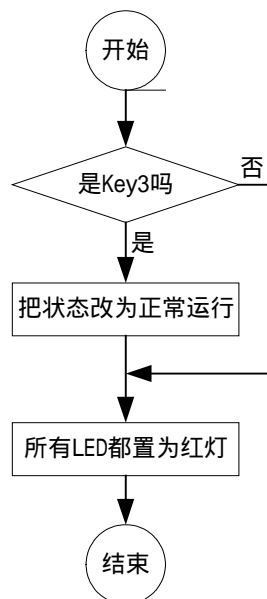


图10 紧急状态子程序

在紧急状态下，所有的 LED 都被置为红灯。在这种状态下，只有 Key3 才可以把状态改为正常运行状态。

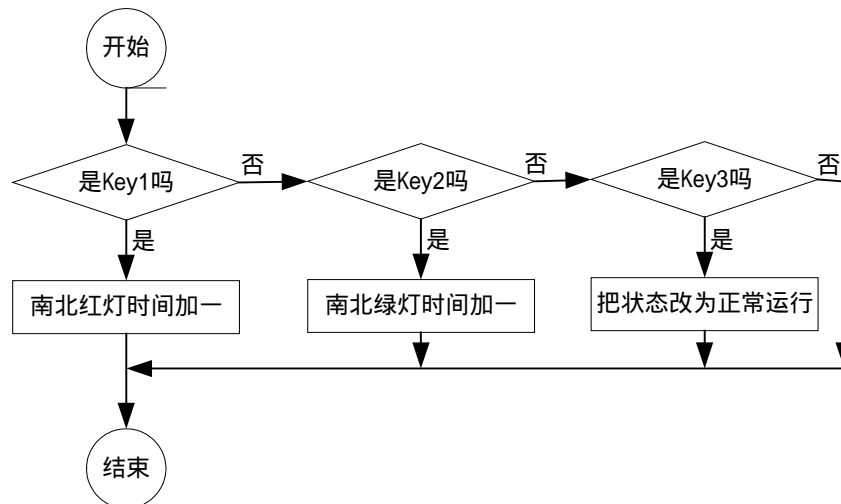


图11 设置状态子程序

在设置状态，按下 Key1 开始设置南北的红灯时间，按下 Key2 设置南北的绿灯时间，按下 Key3 返回正常运行状态。红灯和绿灯的时间最大可以设为 99，超出 99 的时候会从 20 开始重新计数。

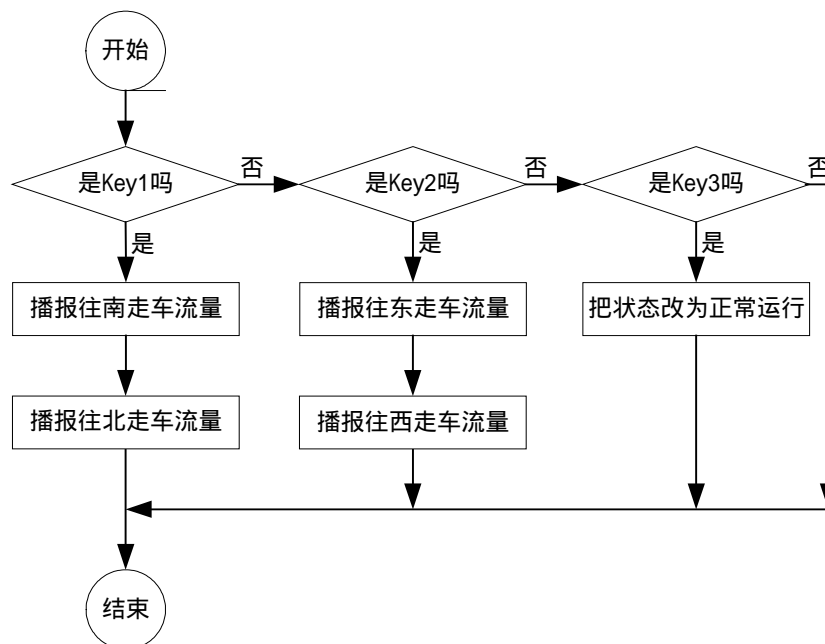


图12 查询状态子程序

在查询状态,按下 Key1 播报南北路口的车流量,按下 Key2 播报东西路口的车流量,按下 Key3 返回正常运行状态。

语音播报采用凌阳公司提供的 SACM-S480。

SACM-S480 压缩算法压缩比较大 80:3, 存储容量大, 适用于语音播放, 如“文曲星”词库。

其相关 API 函数如下所示:

int SACM_S480_Initial(int Init_Index)	//初始化
void SACM_S480_ServiceLoop(void)	//获取语音资料, 填入译码队列
void SACM_S480_Play(int Speech_Index, int Channel, int Ramp_Set)	
void SACM_S480_Stop(void)	//停止播放
void SACM_S480_Pause(void)	//暂停播放
void SACM_S480_Resume(void)	//暂停后恢复
void SACM_S480_Volume(Volume_Index)	//音量的控制
unsigned int SACM_S480_Status(void)	//获取模块的状态
Call F_FIQ_Service_SACM_S480	//中断服务函数

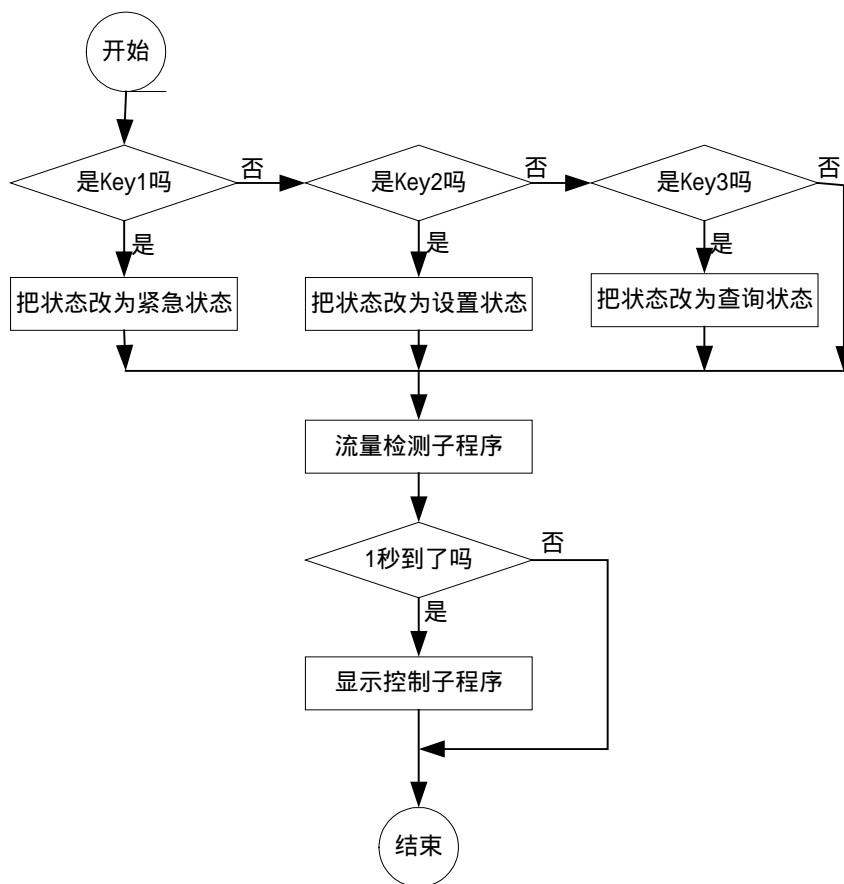


图13 正常运行状态子程序

在正常运行状态下，首先进行按键处理。若按键为 Key1 会进入紧急状态，若按键为 Key2 会进入设置状态，若按键为 Key3 会进入查询状态。然后是流量检测和显示控制。每秒钟去查询子状态，根据子状态的标记去刷新相应的发光二极管和数码管控制单元。流量检测和显示控制的子程序如下面的流程图所示。

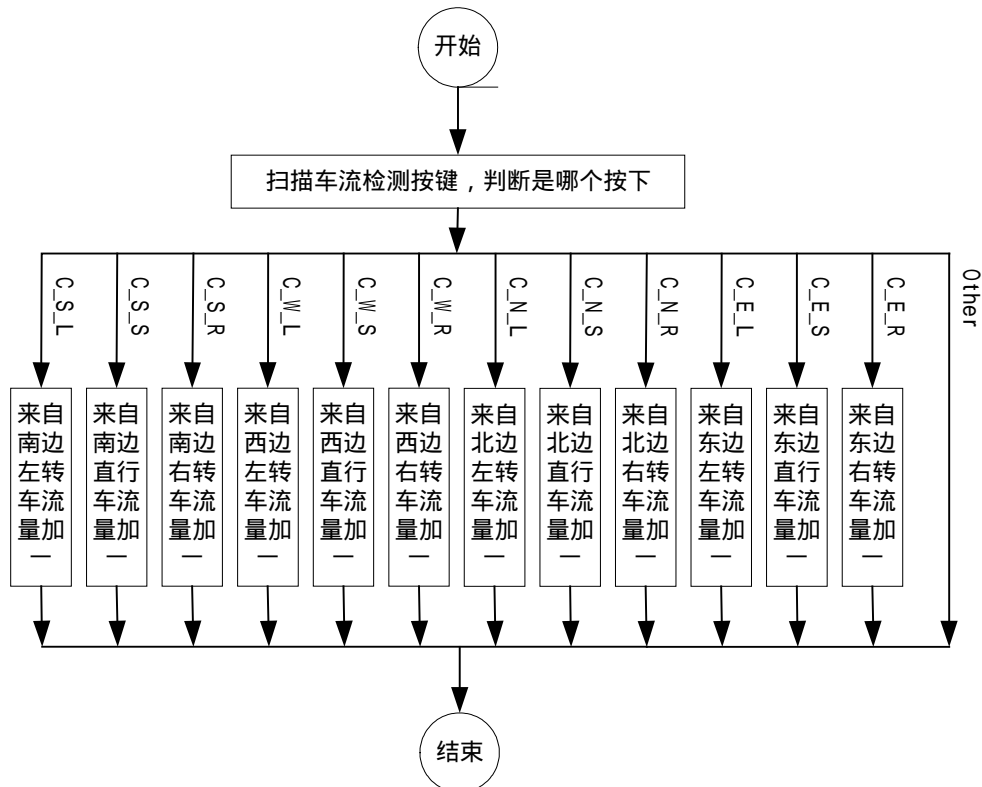


图14 流量检测子程序

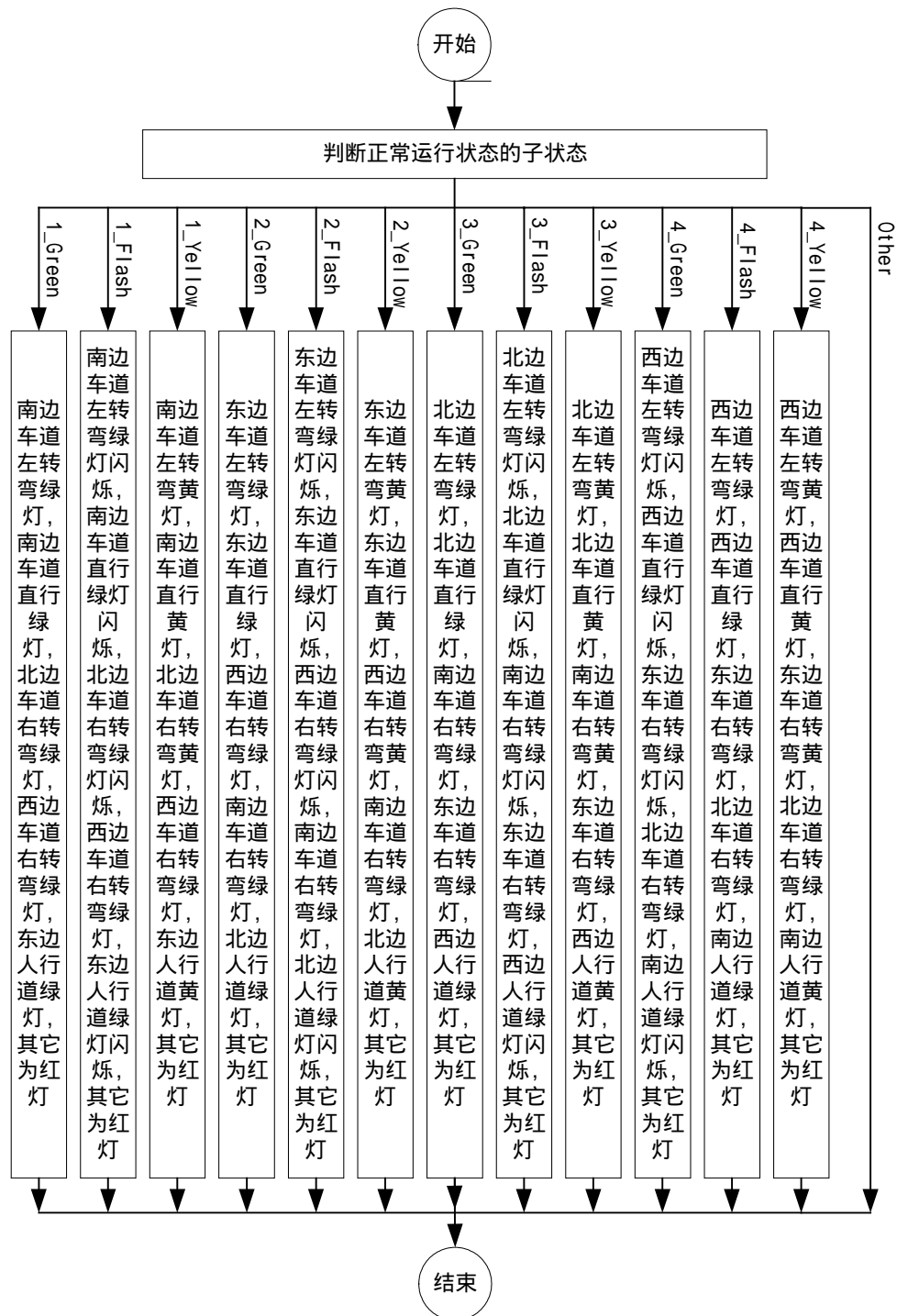


图15 显示控制子程序



