



# 电子系统设计综合实验

## 单片机软件架构设计

-----2023.7

# 目录

CONTENTS

01. 课程概要
02. 设计选题
03. STM32单片机简介
04. 单片机软件架构
05. 编程案例分析



## 第四部分

# 单片机软件架构设计

- 4.1 背景和目标
- 4.2 数据结构
- 4.3 程序架构
- 4.4 测控系统软件规划

## 4.1 背景和目标

- 背景:

- 大家设计软件还缺乏实战经验;
- 软件设计的结构不鲜明, 缺乏条例;

- 目标:

- 增强软件的可读性;
- 使软件的结构清晰, 易于故障定位;
- 规范软件设计的流程;

## 4.2 数据结构（程控交换为例）

- 常用的数据结构：数组、栈和队列

- 数组：是可以再内存中连续存储多个元素的结构，在内存中的分配也是连续的，数组中的元素通过数组下标进行访问，数组下标从0开始。
- 栈：是一种特殊的线性表，仅能在线性表的一端操作，栈顶允许操作，栈底不允许操作。栈的特点是：先进后出，或者说是后进先出，从栈顶放入元素的操作叫入栈，取出元素叫出栈。
- 队列：与栈一样，也是一种线性表，不同的是，队列可以在一端添加元素，在另一端取出元素，也就是：先进先出。从一端放入元素的操作称为入队，取出元素为出队。

- 软件中常用的数据表格：缓冲区，消息队列，用户表（设备表），堆栈，状态表。

# 消息队列

- 信息输入（周期级）程序所得到的用户信息通过消息队列方式传递给基本级程序进行处理；
- 定义：队列是一种操作受限的线性表，对于它所有的插入都在表的一端进行，所有的删除都在表的另一端进行。
- 是一种先进先出(FIFO)的线性表；
- 可分为顺序队列、循环队列和链形队列，单片机中，一般采用循环队列；

# 消息队列

## 消息队列工作过程-1

地址序号	内容
1	设备号1
2	信息号1
信息读出指针（队首） →	设备号2
4	信息号2
信息写入指针（队尾） →	设备号3
5	设备号3
6	信息号3
...	...
255	设备号128
256	信息号128

注： 每个信息包由两字节组成，设备号在前，信息号在后。

信息号： 0~15：表示号码0, 1, 2。。。9, \*, #, A, B, C, D；

16：表示 摘机；

17：表示 挂机；

18：表示 时间到；

19：表示 ZG振铃；

20：表示 收到FSK，作FSK纠错译码处理；

21：表示 FSK发送完毕，作下一步处理；

22：表示 RS232收到数据包。

每条消息占用2字节，前一字节表示消息的来源——设备号；后一字节表示消息的类型或内容——消息号

## 消息队列工作过程-2

- 每当周期级程序得到一条消息就向消息队列写入该条消息，队尾指针递增2字节；
- 基本级程序则一直在比较队尾和队首的指针：若两者相等，表明没有待处理的消息；若两者不等，表明队列中有未处理消息，程序将队首所指的取出进行处理，同时把队首指针递增2字节，待消息处理完后再进行队首和队尾指针的比较，直到队首队尾指针相等。



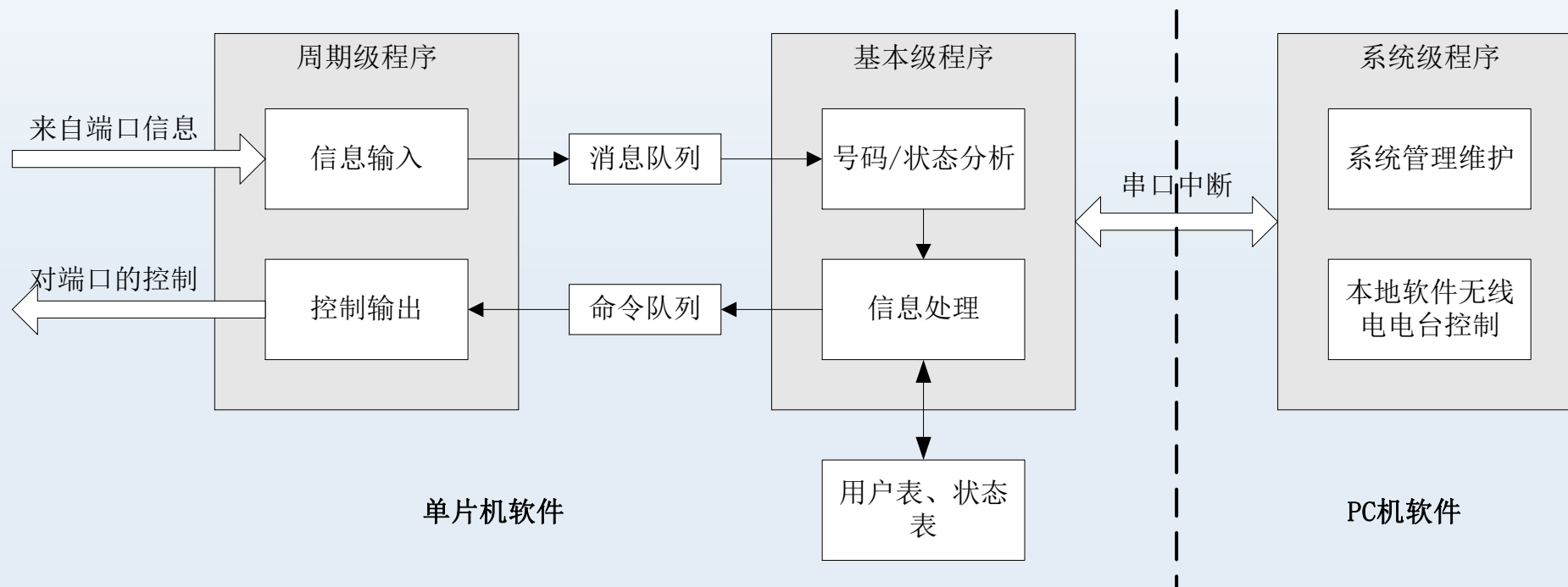
# 输出数据表

- 输出数据表主要用于基本级程序向周期级程序发送的输出控制消息；
- 依据输出控制的类型不同，可将输出数据表分为多个表。  
如：振铃数据表、话路表和用户音调表等；
- 周期级程序，周期性地监测输出数据表的内容，有数值变化时，对端口进行控制输出。

# 接收/发送缓冲区

- 一般用于一些数据收发的暂存。
  - 如在串行通信中，可设置接收和发送两个缓冲区 (scomm\_rev\_buff, scomm\_send\_buff)；
- 表征缓冲区的参数：
  - 缓冲区首址（常数）；
  - 缓冲区长度（常数）；
  - 计数器（变量）；
- 一般设为全局变量。

## 4.3 程序架构框图



## 4.3 程序架构

- **周期级（信息扫描与输出）程序：**

主要用于收集系统来自各设备端口以及系统总体的信息，如：有线用户的摘/挂机、有线用户的拨号、中继端口的振铃、无线端口的FSK拨号、超时信息、RS232串口信息等等。除RS232串口信息是通过串口中断实现以外，其他的信息都是在**定时中断**程序内完成的。

- **基本级（信息处理）程序：**

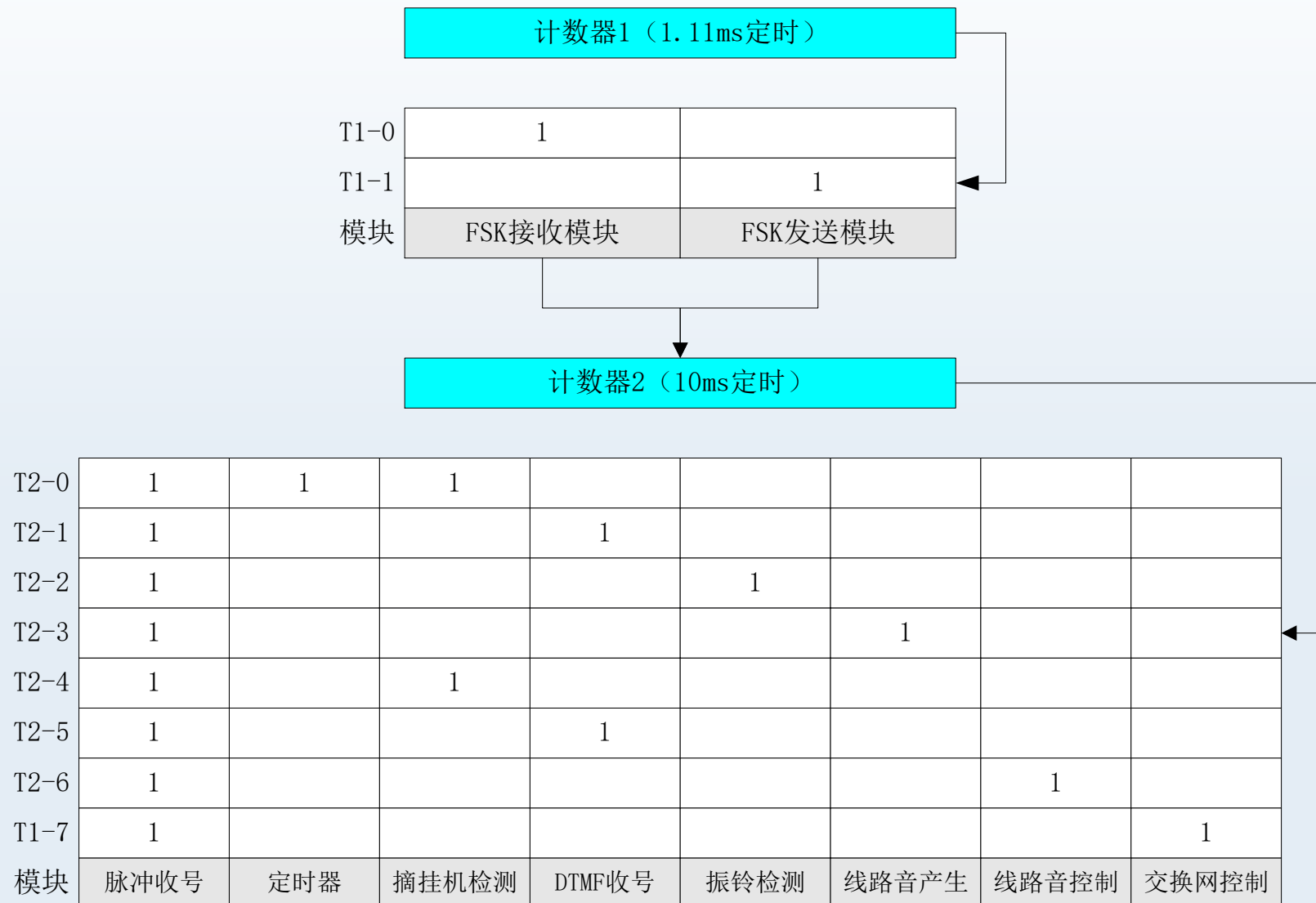
是系统中最为复杂的部分，由主程序来完成。信息的处理要依据信息本身、信息来自的设备端口以及该端口的状态等综多因素来进行。由于设备端口的类型各不相同，同时，每一类端口的状态又多种多样，所以在信息处理程序设计时，必定采用**状态迁移图**的方式进行。

- **系统级（系统管理维护）程序：**

由单片机和PC机联合完成。主要实现用户的权限管理，对监听端口及其音量的控制、强行建链/拆链的控制、人工转接控制等等。PC机主要实现控制管理，单片机是具体执行者。

# 周期级软件设计

- 周期级软件用于周期地收集系统来自各设备端口以及系统本身的信息（如：有线用户的摘/挂机状态、有线用户的拨号、中继端口的振铃、无线端口的FSK拨号、定时信息等等），生成各种事件队列供基本级软件接收处理；
- 接收来自基本级软件的命令（如：中继占用、交换网络的控制、FSK拨号输出、信号音输出等等），去控制相应的端口。
- 位处理技术（均衡任务）



# 其它中断程序

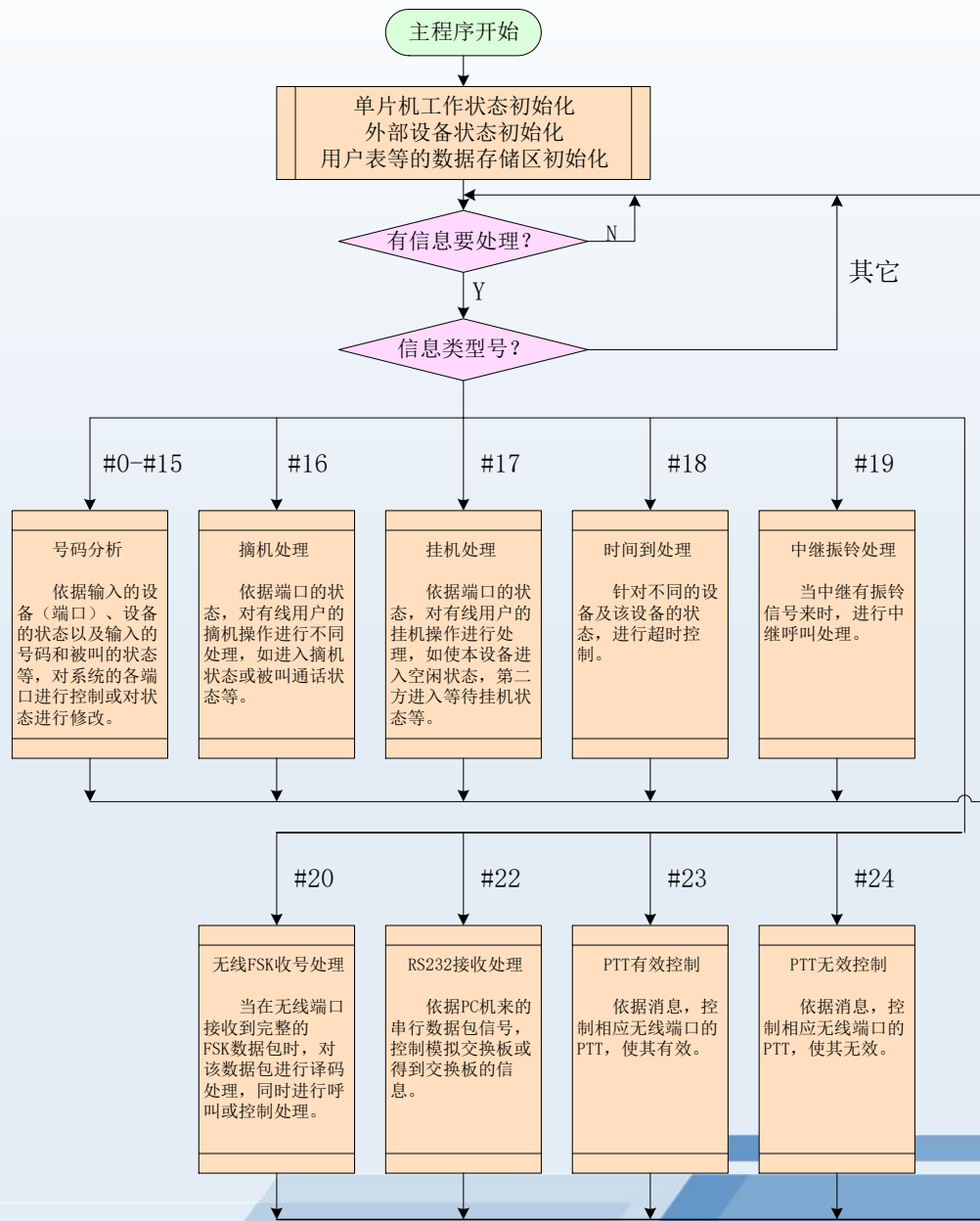
- 外部中断
- 串行中断
- ADC/DAC中断

# 基本级软件设计

- 是系统中最为复杂的部分，由主程序来完成。
  - 信息的处理要依据信息本身、信息来自的设备端口以及该端口的状态等众多因素来进行。
  - 由于设备端口的类型各不相同，同时，每一类端口的状态又多种多样，所以在信息处理程序设计时，必定采用状态迁移图的方式进行。
- 当前状态 + 激励事件 = 新状态

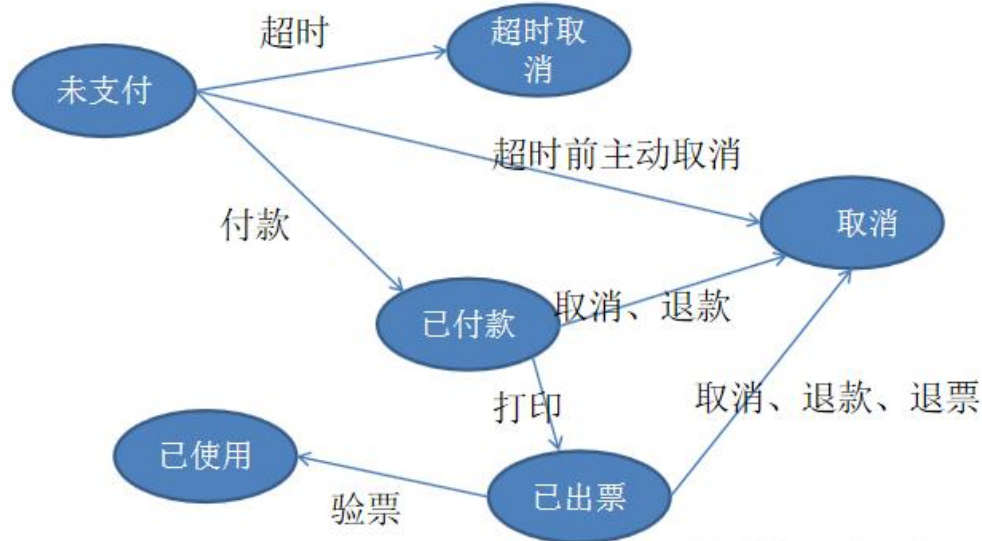


# 基本级软件的总体流程图

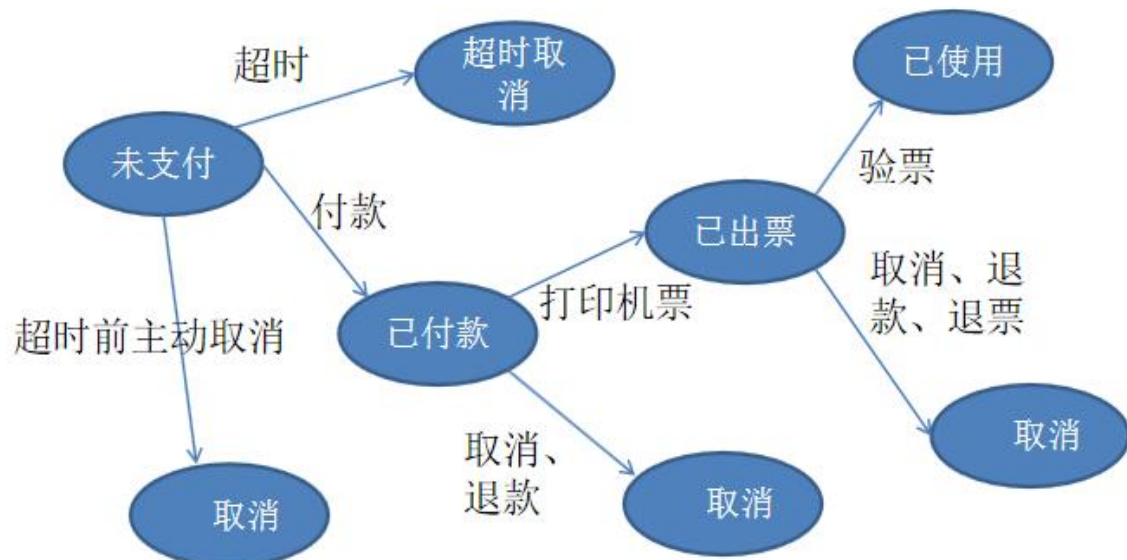


# 状态迁移图和状态转换树

• 当前状态 + 激励事件 = 新状态



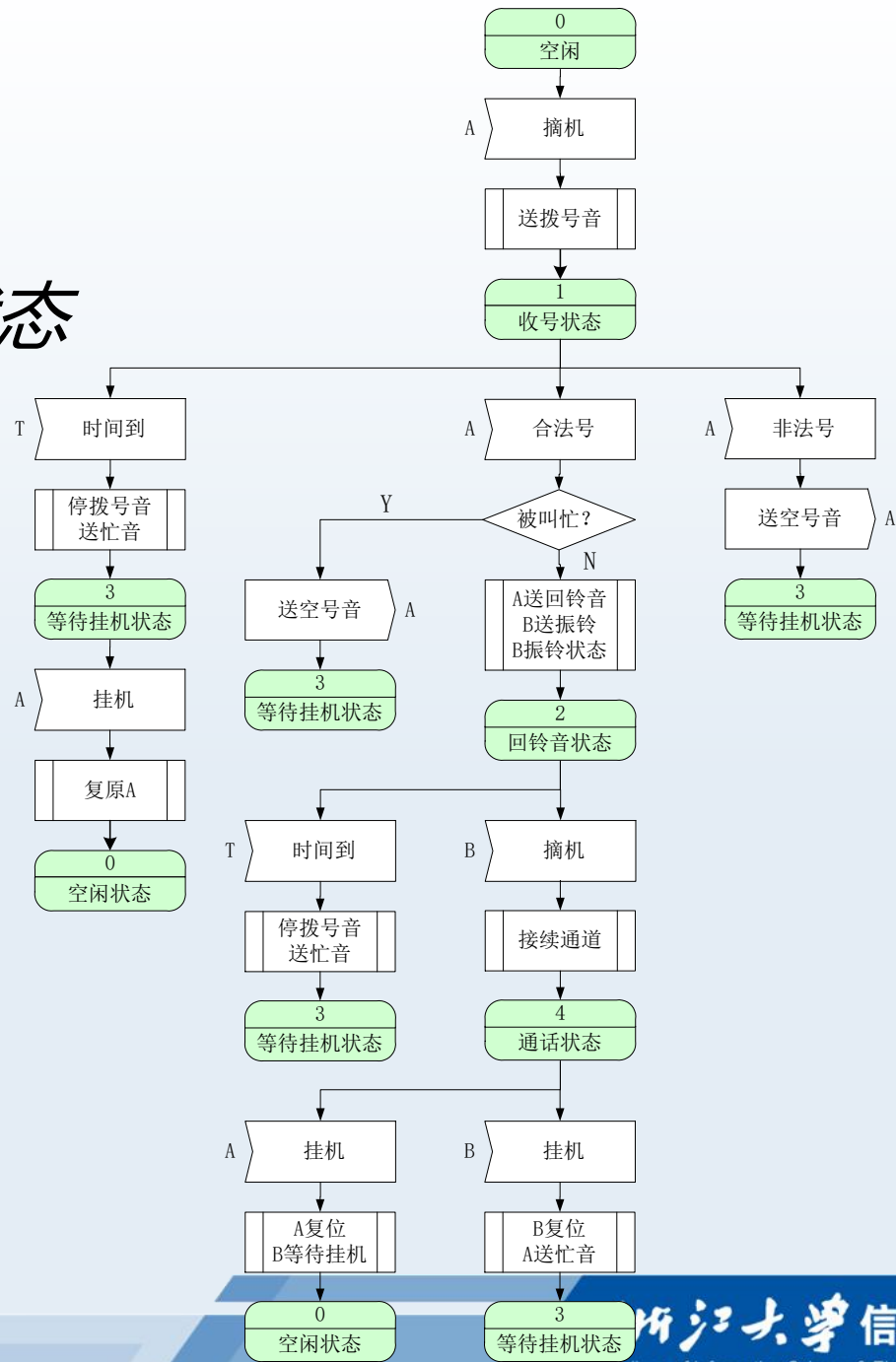
<https://blog.csdn.net/xinanbo>



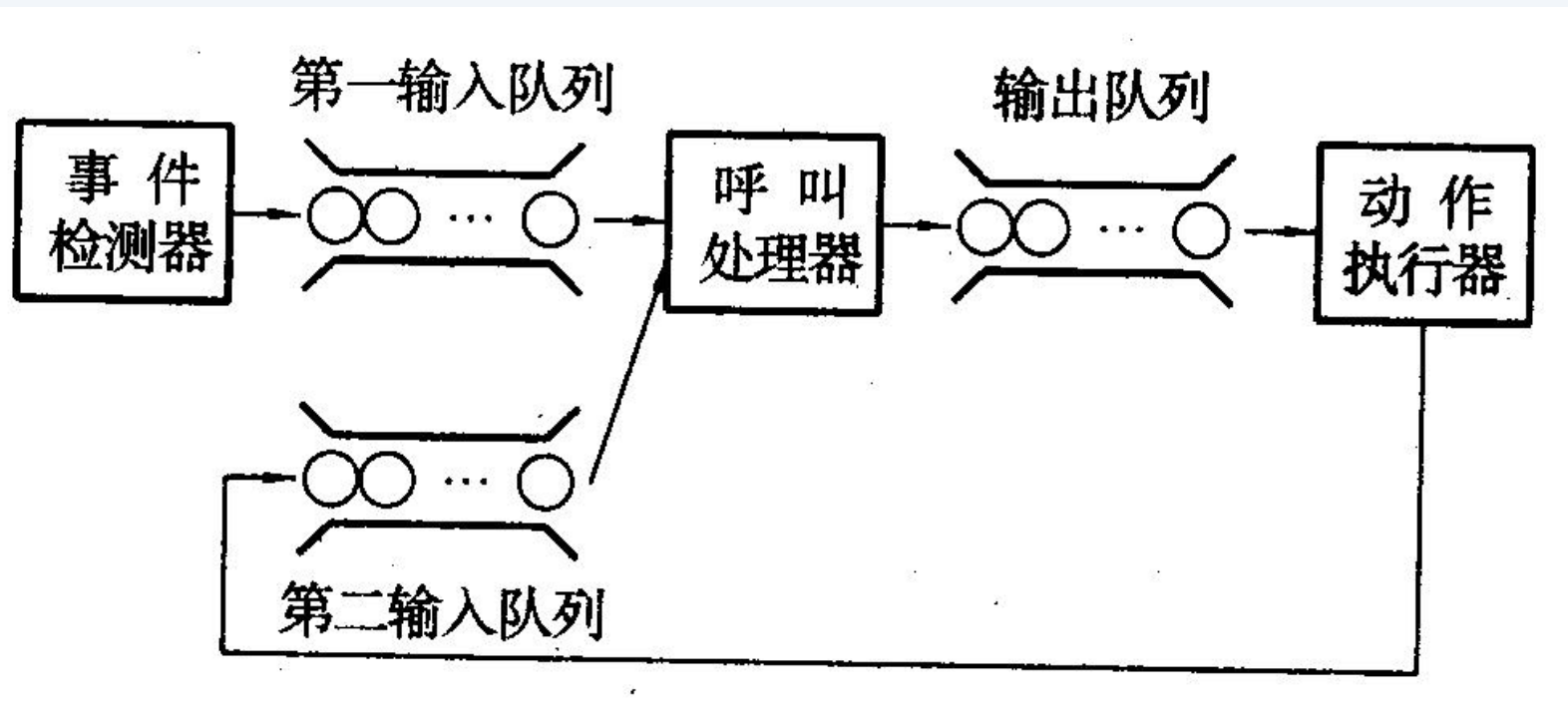
<https://blog.csdn.net/xinanbo>

# 程控交换状态转换树

- 当前状态 + 激励事件 = 新状态



# 基本级软件-消息驱动



# 系统级程序设计

- 由单片机和PC机（上位机）联合完成。主要实现用户的权限管理，强行建链/拆链的控制、人工转接控制等等。PC机主要实现控制管理，单片机是具体执行者。
- 用户参数和用户权限管理；
- 也可通过单片机软件的人机界面实现，如：通过人机界面设置系统的参数等。

## 4.4 测控系统软件设计（温控项目为例）

### • 周期级

- 按键扫描；
- 定时采样（即启动ADC）；
- 查询或中断方式取得ADC数据，存于变量或数组（可用于平滑处理）。

### • 基本级

- AD值的平滑处理（去干扰）；
- 温度值的折线拟合；
- 温度的PID控制算法实现；
- 状态参数的OLED显示。

### • 系统级

- 设置温控范围，设定PID控制系数；
- 通过串口发送温控曲线；

- 涉及到的设备接口
  - 定时器（中断方式，启动ADC，检测按键，检测端口状态）
  - GPIO（按键，LED状态指示）
  - ADC（中断，或查询方式）
  - PWM（温度控制）
  - Uart接口（接收采用中断方式，发送采用中断，或查询方式）



The End