

自角度传感器的两路角度信号 PITCH 和 ROLL, 用来感知飞行器的当前姿态并用来与目标姿态进行比较, 实现闭环控制; 来自三个单轴角速率陀螺的三路角速率信号, 用来增加阻尼, 提高飞行器的飞行稳定性。微处理器将所获得的这 9 路信号之后, 将这些信号通过 PID、及分段比例控制运算, 计算得出四个旋翼的控制量, 并以 PWM 占空比的形式分别通过端口 P0.4、P0.5、P0.6、P0.7 输出控制旋翼动作。

控制系统的控制运算如图 3-30 所示, 将各传感器所获得的信号与遥控指令相比较, 得出各个偏差 e , 通过 PID 控制算法得出各传感信号对系统的控制量, 然后再将这些控制量根据控制规则进行分段比例控制, 得出最终的控制量, 并以 PWM 的形式输出控制各个旋翼。

3.3.2 系统初始化

由于系统需要一个精确的时基信号, 因此考虑采用 24M 的外部晶体。系统初始化程序如下:

```
//初始化系统时钟, 使用外部振荡器
void SYSCLK_Init(void)
{
    int i;                //延时计数器
    OSCXCN=0x67;         //启动外部振荡器
    for (i=0;i<256;i++); //等待振荡器起振
    while(!(OSCXCN&0x80)); //等待晶体振荡器稳定
    OSCICN=0x88;         //选择外部振荡器作为系统时钟源
}
```

3.3.3 交叉开关和 I/O 口配置

C8051F021 具有丰富的内部资源, 可以通过优先权交叉开关译码器(即交叉开关)将端口 0~3 的引脚分配给器件上的数字外设。分配顺序是从 P0.0 开始, 可以一直分配到 P3.7。

本设计将 UART0、UART1、PCA 等内部资源分配给各端口引脚。引脚配置步骤如下:

● 交叉开关引脚分配:

当交叉开关配置寄存器 XBR0, XBR1 和 XBR2 中外设的对应允许位被设置为逻辑 1 时, 交叉开关将端口引脚分配给外设, 特殊功能寄存器的设置如下所示。

特殊功能寄存器 XBR0 的设置:

| | | | | | | | |
|------|-------|--------|-----|-----|---------|--------|--------|
| 位 7 | 位 6 | 位 5 | 位 4 | 位 3 | 位 2 | 位 1 | 位 0 |
| CP0E | ECIOE | PCA0ME | | | UART0EN | SPI0EN | SMB0EN |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |

特殊功能寄存器 XBR1 的设置:

| | | | | | | | |
|--------|-------|-----|-------|-----|-------|-----|------|
| 位 7 | 位 6 | 位 5 | 位 4 | 位 3 | 位 2 | 位 1 | 位 0 |
| SYSCKE | T2EXE | T2E | INT1E | T1E | INT0E | T0E | CP1E |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

特殊功能寄存器 XBR2 的设置:

| | | | | | | | |
|---------|-------|-----|-------|-----|--------|--------|--------|
| 位 7 | 位 6 | 位 5 | 位 4 | 位 3 | 位 2 | 位 1 | 位 0 |
| WEAKPUD | XBARE | -- | T4EXE | T4E | UART1E | EMIFLE | CNVSTE |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

UART0 有最高的优先权, 则 P0.0 被分配给 TX0, P0.1 被分配给 RX0;

UART1 的优先权次之, 则 P0.2 被分配给 TX1, P0.3 被分配给 RX1;

然后是 PCA, 所以 CEX0 被分配给 P0.4, CEX1 被分配给 P0.5, CEX2 被分配给 P0.6, CEX3 被分配给 P0.7;

● 配置端口引脚的输出方式:

每个端口引脚的输出方式都可被设置为漏极开路或推挽方式。端口 0~3 引脚的输出方式由 PnMDOUT 寄存器中的对应位决定, 当对应位为逻辑 1 时, 对应端口配置为推挽方式, 为逻辑 0 时, 对应端口配置为漏极开路方式, 所有端口引脚的缺省方式均为漏极开路。

端口 0 输出方式寄存器 P0MDOUT 的设置:

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 位 7 | 位 6 | 位 5 | 位 4 | 位 3 | 位 2 | 位 1 | 位 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

0 端口引脚的输出方式为漏极开路;

1 端口引脚的输出方式为推挽方式;

● 程序实现

```
void PORT_Init(void) //配置交叉开关和 GPIO 端口
{
    XBR0=0x24; //将 TX0 连到 P0.0, RX0 连到 P0.1, CEX0..CEX3 连到 P0.4..P0.7
    XBR1=0x00;
    XBR2=0x44; //允许交叉开关和弱上拉
    P0MDOUT|=0XF0; //允许 P0.4..P0.7 为推挽输出
}
```