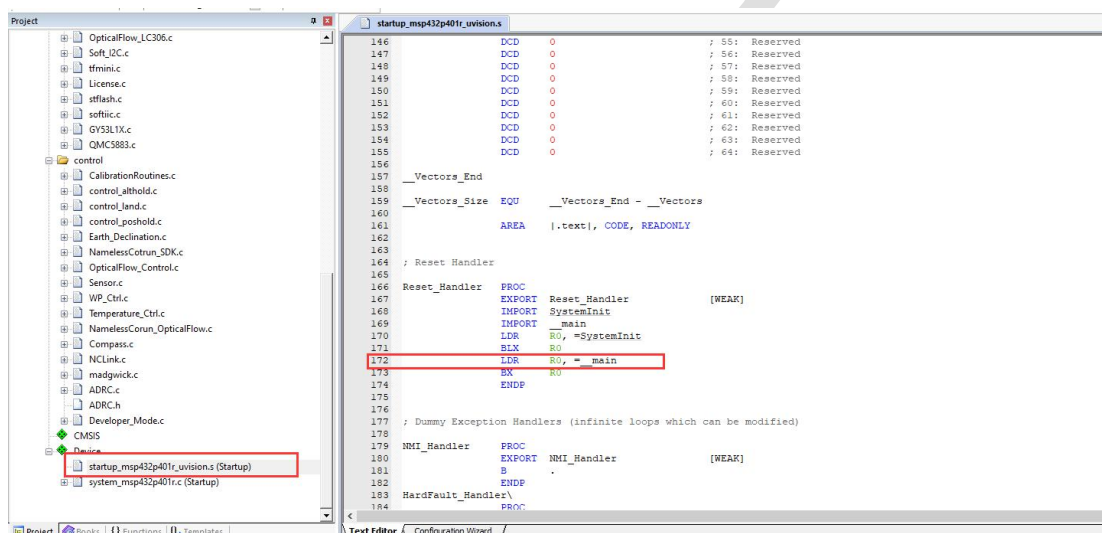


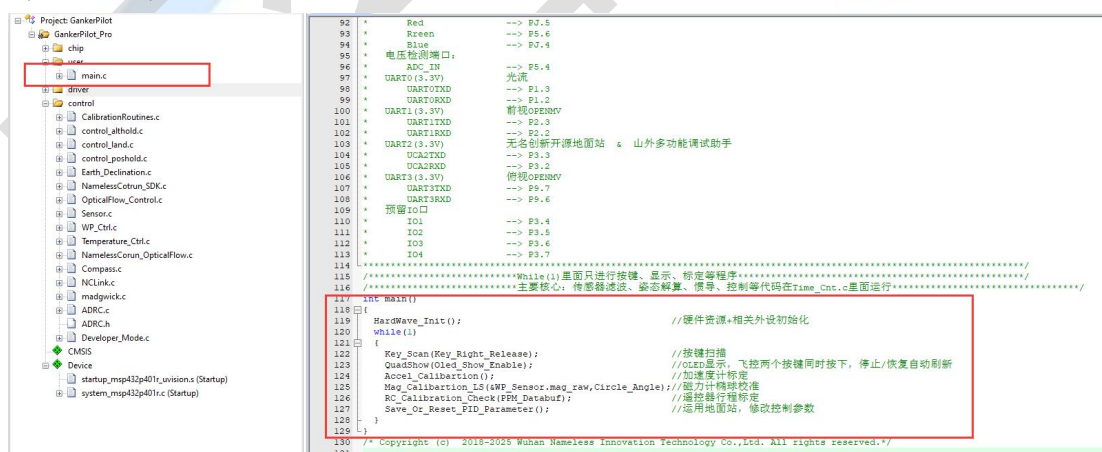
六、MSP432 飞控软件框架分析

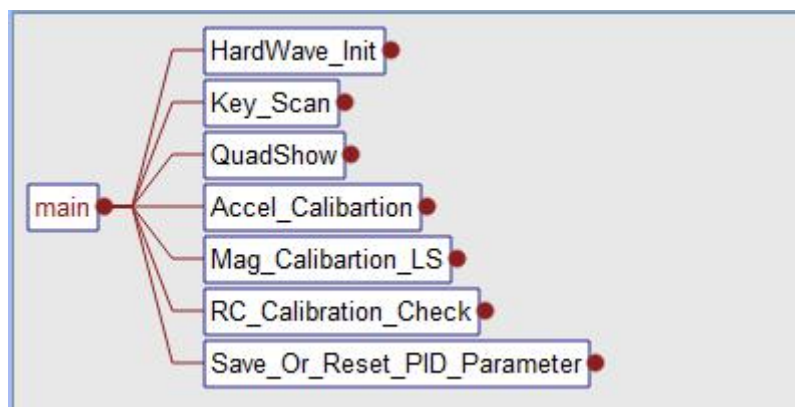
飞控整个软件代码运行在定时器调度模式，同时对各个外设模块中断优先级进行了合理设计，确保程序运行的实时性和稳健性。其中对实时性、周期性要求高的比如：传感器数据采集、姿态解算、惯性导航、控制等任务，采用定时器调度的方式执行，其它耗时且对周期性要求不高的子任务例如：电压采集、按键扫描、显示屏刷新、地面站发送、加速度计标定、磁力计标定、遥控器行程标定、参数保存等放在 while(1) 里面，利用中断空闲的时间去执行。

飞控上电复位后，单片机会首先根据官方启动文件定义来初始化堆栈指针、程序计数器、堆栈大小、中断向量表等一系列过程，最后会运行 C 函数分支入口 _main，startup_msp432p401r_uvision.s 内容仅做了解就可以。

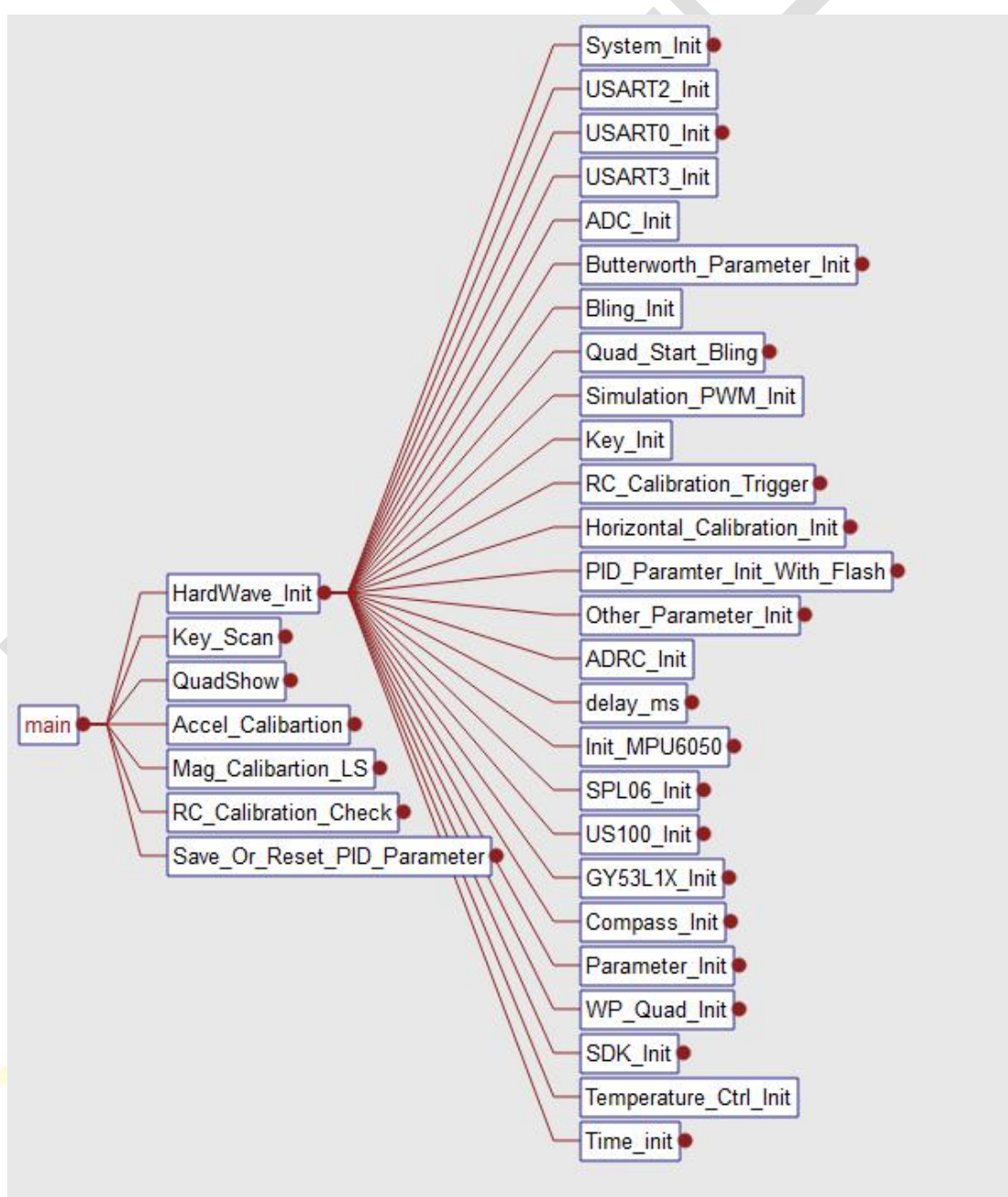


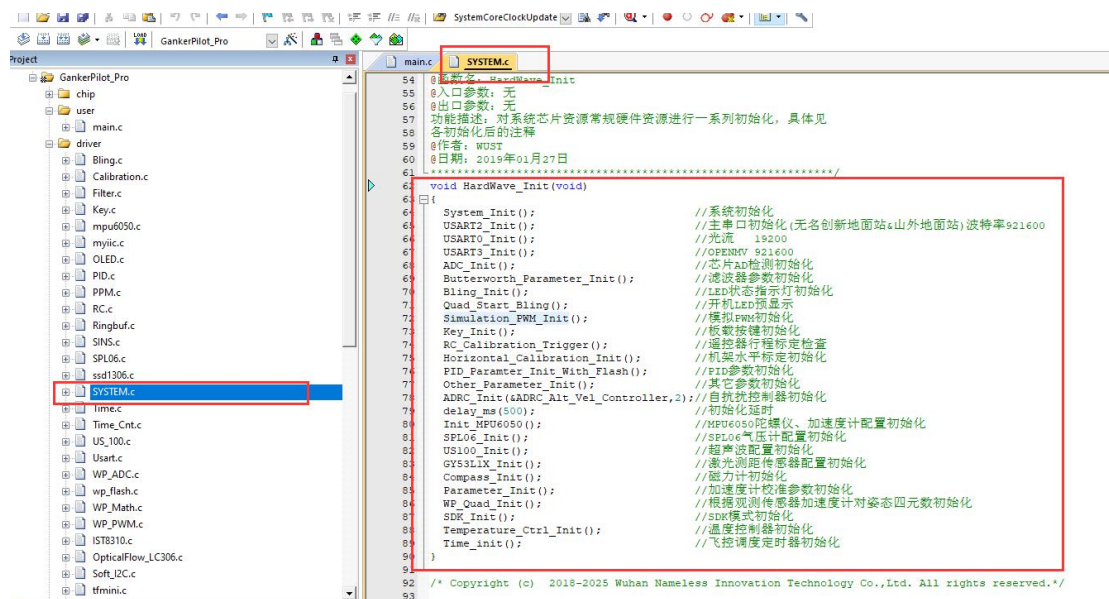
下面我们利用 Understand 软件来对飞控程序框架进行分析，首先看 main 主函数的调用框图：



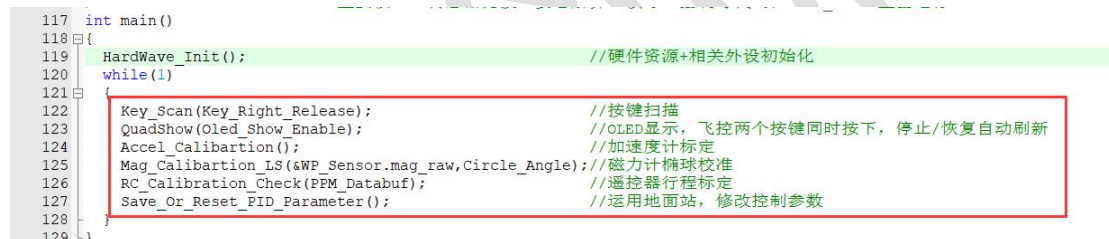


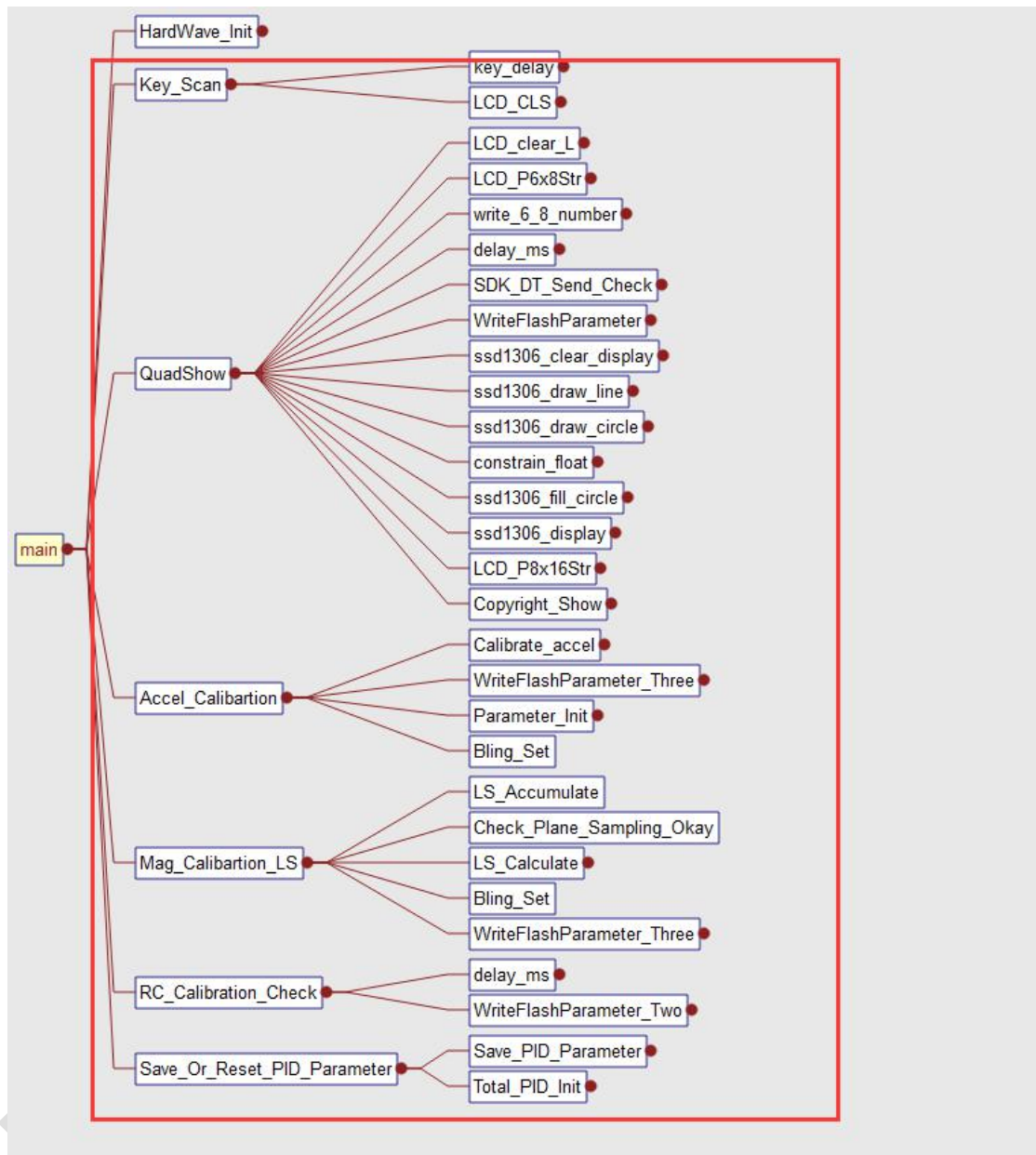
首先主函数 `main` 会调用硬件初始化函数 `HardWave_Init()`，硬件初始化包括以下内容：





主函数 main 执行硬件初始化函数 HardWave_Init() 完毕后, 会进入 while(1) 内利用中断空闲时间执行按键扫描、OLED 显示、加速度计校准、磁力计校准、遥控器行程校准、参数保存等。

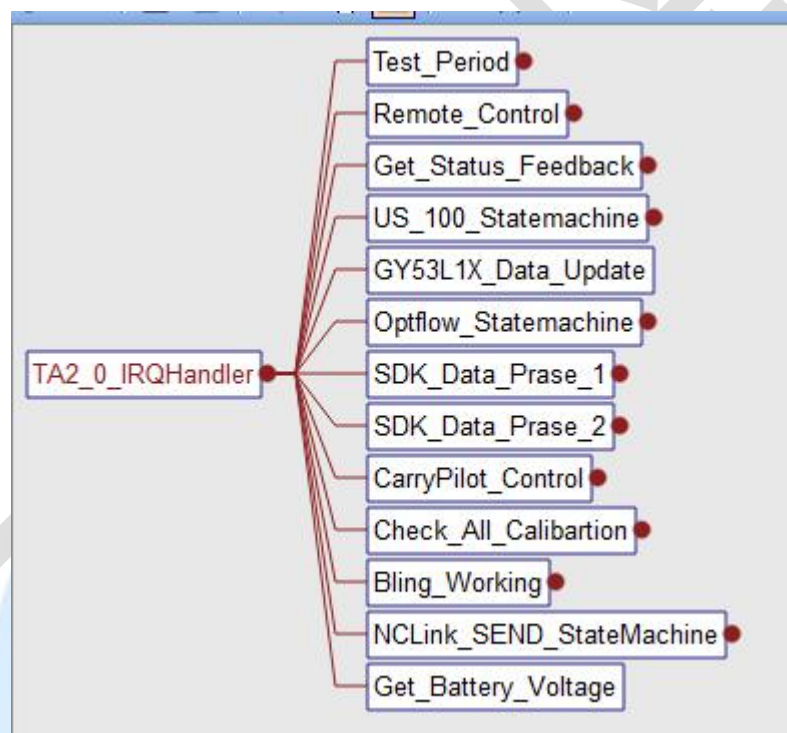


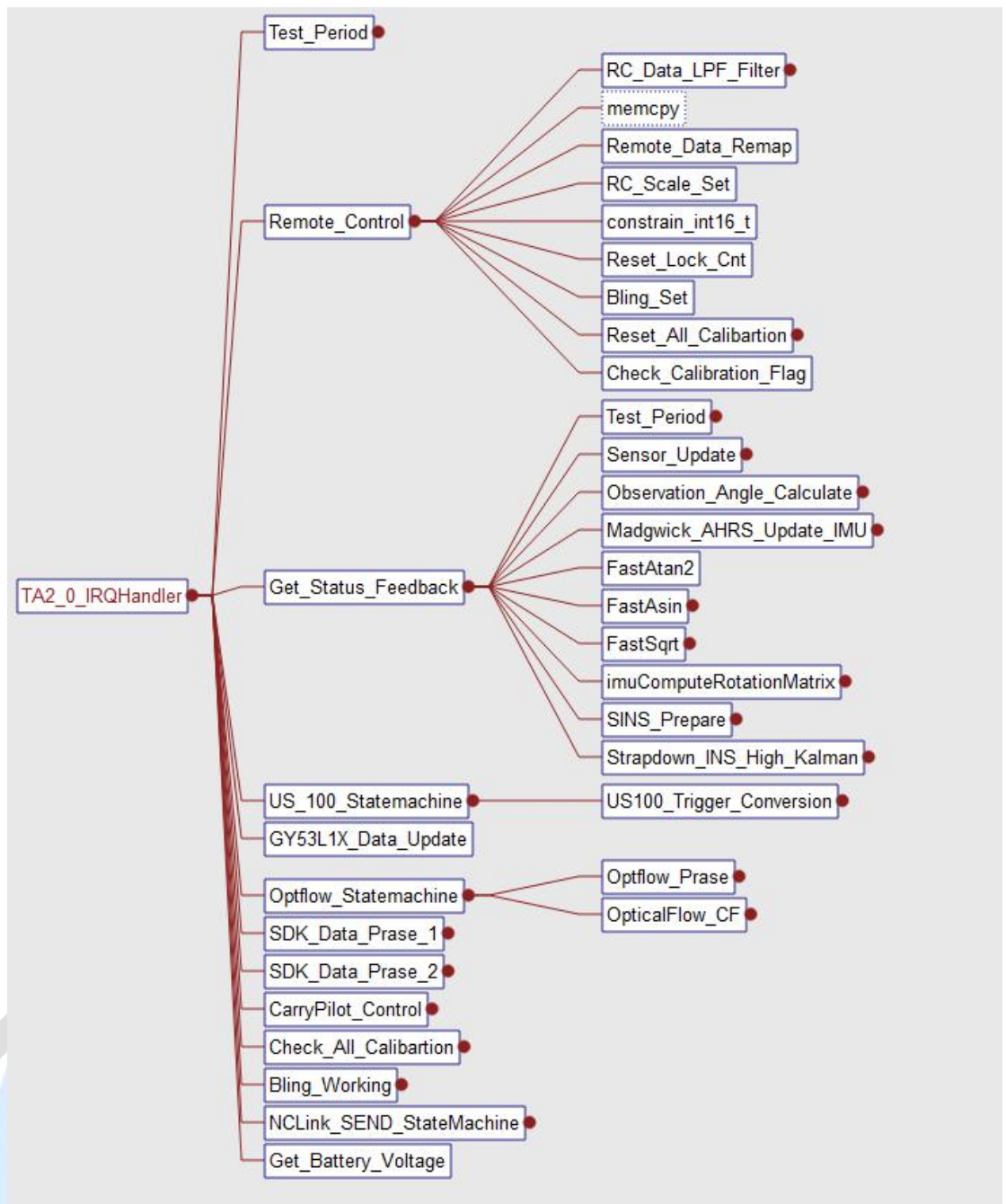


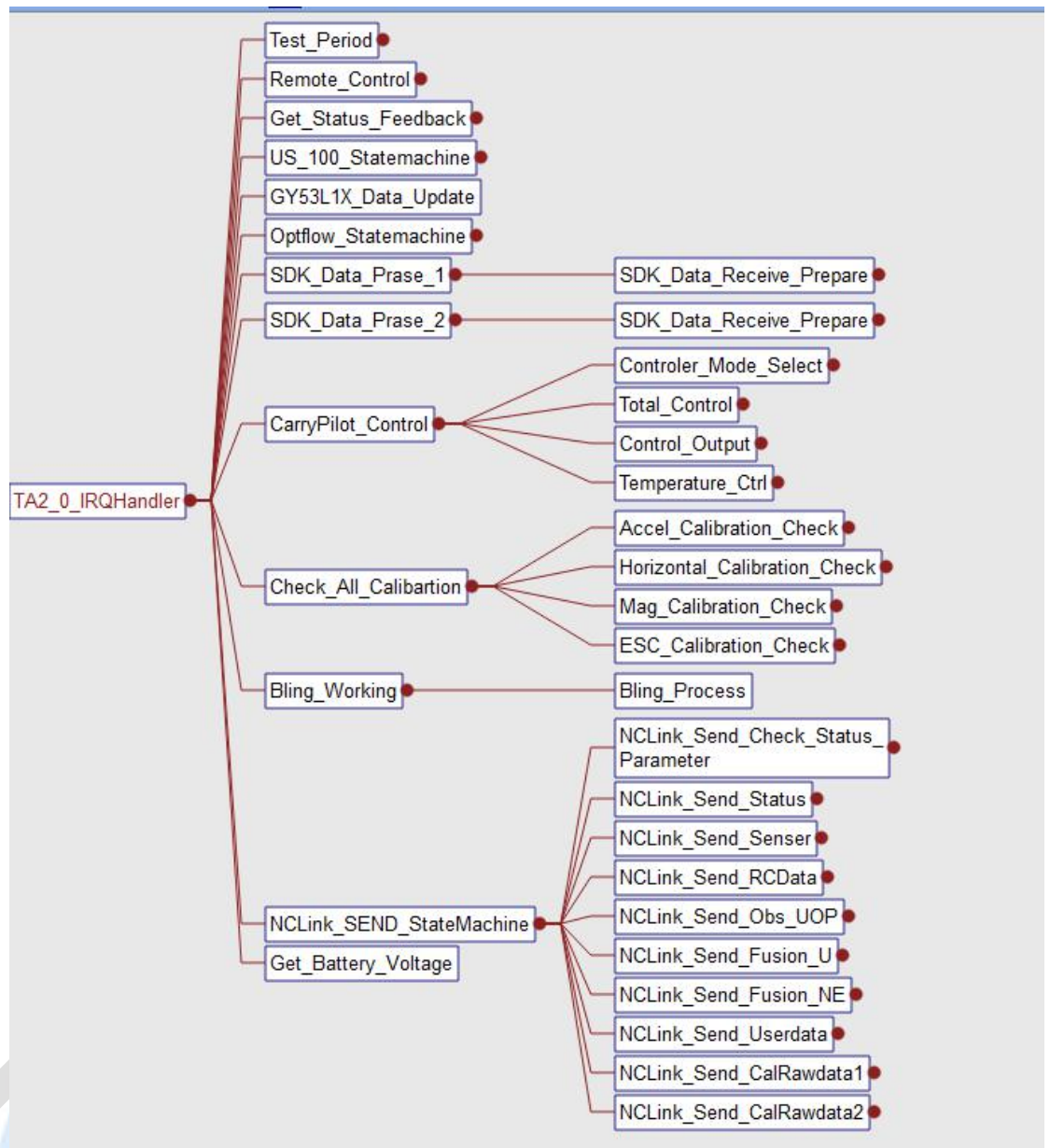
定时器中断任务调度主要执行的任务为遥控器数据转换、传感器数据采集、姿态解算、光流速度融合、惯导 GPS 数据融合、SDK 数据解析、控制、校准检测等。


```
Project: GankePlot
main.c SYSTEM.c Time_Cnt.c
chip
user
main.c
driver
Bling.c
Calibration.c
Kaz.c
myu6050.c
mytic.c
OLED.c
PID.c
PPM.c
RC.c
Ringbuf.c
SMB.c
SPD.c
ssd1306.c
SYSTEM.c
Time.c
Time_Cnt.c
US_100.c
Uart.c
WP_ADC.c
wp_flash.c
WP_Math.c
WP_PWM.c
IST8310.c
OpticalFlow_LC306.c
Soft_I2C.c
License.c
stflsh.c
softiic.c
GY53L1X.c
QMC5883.c
control
CalibrationRoutine.c
control_alhoid.c
control_lamd.c
control_spyhold.c

117
118 Sensor WP_Sensor;
119 Testtime Time0_Delta;
120 Testtime Time0_Delta;
121 float test_time;
122 /*****
123 函数名: TA2_0_IRQHandler
124 入口参数: 无
125 出口参数: 无
126 功能描述: 系统调度定时器中断服务函数, 主要进行遥控器解析、
127 传感器数据采集、数字滤波、姿态解算、惯性导航、控制等对周期有
128 严格要求的函数
129 作者: WUST
130 日期: 2019年01月27日
131 *****/
132 void TA2_0_IRQHandler(void)//5ms刷新一次
133 {
134     Test_Period(&Time0_Delta);
135     Remote_Control(); //遥控器数据解析
136     Get_Status_Feedback(); //获取姿态数据、水平与垂直方向惯导数据
137     #if (Ground_Distance_Sensor==US100) //超声波传感器状态机更新
138     US_100_StateMachine();
139     #else
140     TFmini_StateMachine(); //激光TFMINI_PLUS传感器状态机更新
141     #endif
142     GY53L1X_Data_Update();
143     Optflow_StateMachine(); //光流状态机, 初始化时存在光流外设
144     SDK_Data_Prase_1(); //SDK数据解析
145     SDK_Data_Prase_2();
146     CarryPilot_Control(); //总控制器
147     Check_All_Calibartion();
148     Bling_Working(Bling_Mode); //状态指示灯运行
149     NCLink_SEND_StateMachine(); //无名创新开源地面站, 下载链接http://nameless.tech/download.html
150     Get_Battery_Voltage(); //测量电池电压
151     Test_Period(&Time0_Delta);
152     if(Time0_Delta.Now_Time-Time0_Delta.Now_Time>test_time) test_time=Time0_Delta.Now_Time-Time0_Delta.Now_Time;
153     MAP_Timer_A_clearCaptureCompareInterrupt(TIMER_A2_BASE,TIMER_A_CAPTURECOMPARE_REGISTER_0);
154     // VCan_Send(); //山外地面站 <多功能调试助手>
155 }
156
157
158 /* Copyright (c) 2018-2025 Wuhan Nameless Innovation Technology Co.,Ltd. All rights reserved.*/
159
160
161
```





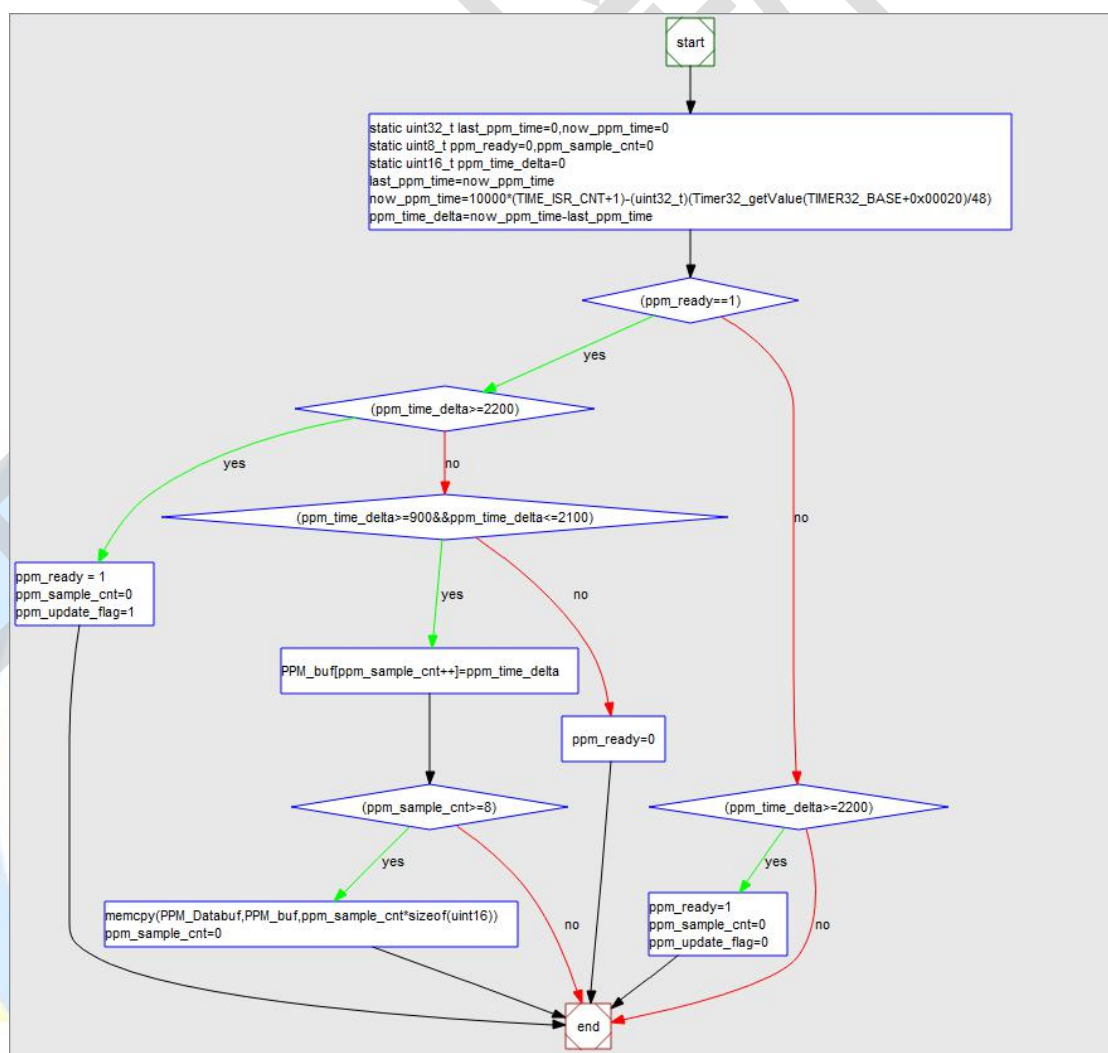


其它中断服务函数如 PPM 中断解析函数、串口数据解析函数，当外设模块发送中断信号后，会进入对应中断函数：

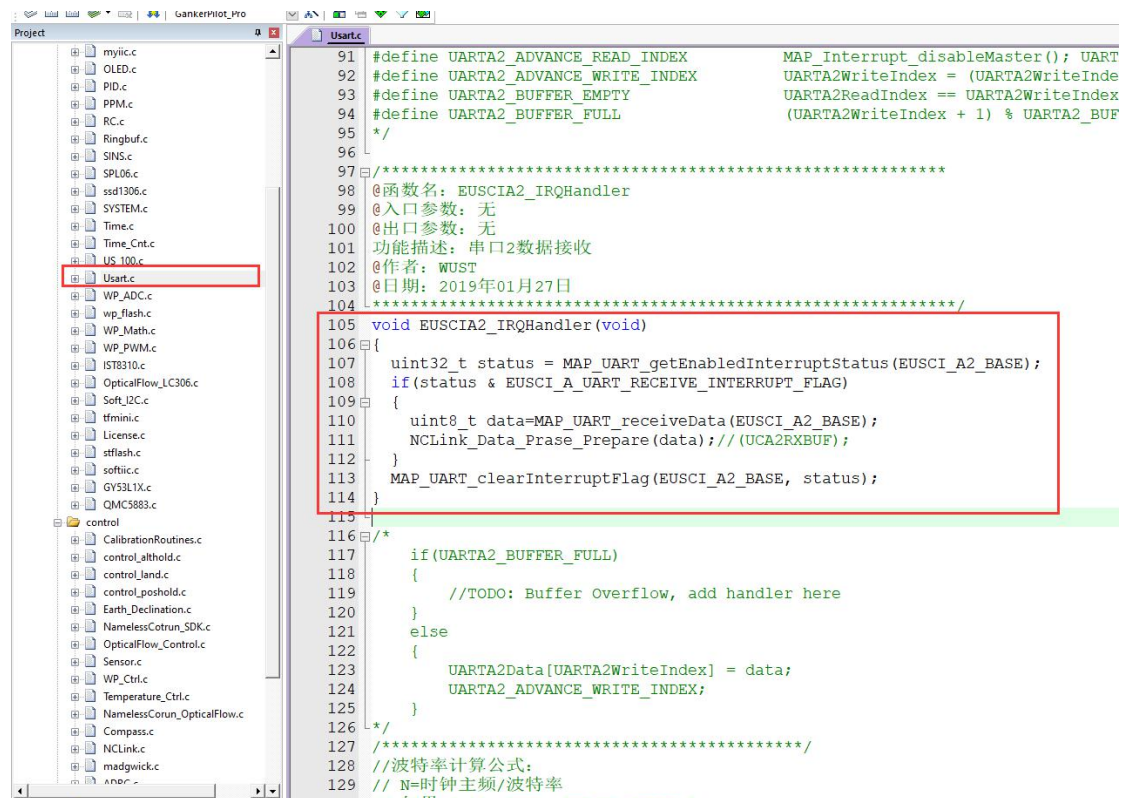
```

73 void PPM_IRQHandler(void)
74 {
75     static uint32_t last_ppm_time=0,now_ppm_time=0;
76     static uint8_t ppm_ready=0,ppm_sample_cnt=0;
77     static uint16_t ppm_time_delta=0;
78     last_ppm_time=now_ppm_time;//系统运行时间获取,单位us
79     now_ppm_time=10000*(TIME_ISR_CNT+1)-(uint32_t)(Timer32_getValue(TIMER32_BASE+0x00020)/48);//单位ms//us
80     //now_ppm_time=micros();
81     ppm_time_delta=now_ppm_time-last_ppm_time;
82     //PPM解析开始
83     if(ppm_ready==1)
84     {
85         if(ppm_time_delta>=2200)//帧结束电平至少2ms=2000us,由于部分老版本遥控器、
86         //接收机输出PPM信号不标准,当出现解析异常时,尝试改小此值,该情况仅出现一例:使用天地飞老版本遥控器
87         {
88             //memcpy(PPM_Databuf,PPM_buf,ppm_sample_cnt*sizeof(uint16));
89             ppm_ready = 1;
90             ppm_sample_cnt=0;
91             ppm_update_flag=1;
92         }
93         else if(ppm_time_delta>=900&&ppm_time_delta<=2100)
94         {
95             PPM_buf[ppm_sample_cnt++]=ppm_time_delta;//对应通道写入缓冲区
96             if(ppm_sample_cnt>=8)//单次解析结束
97             {
98                 memcpy(PPM_Databuf,PPM_buf,ppm_sample_cnt*sizeof(uint16));
99                 //ppm_ready=0;
100                 ppm_sample_cnt=0;
101             }
102         }
103         else ppm_ready=0;
104     }
105     else if(ppm_time_delta>=2200)//帧结束电平至少2ms=2000us
106     {
107         ppm_ready=1;
108         ppm_sample_cnt=0;
109         ppm_update_flag=0;
110     }
111 }
112
113
114

```



PPM 数据解析



```
91 #define UARTA2_ADVANCE_READ_INDEX      MAP_Interru...
92 #define UARTA2_ADVANCE_WRITE_INDEX     UARTA2WriteIndex = (UARTA2WriteInde
93 #define UARTA2_BUFFER_EMPTY            UARTA2ReadIndex == UARTA2WriteIndex
94 #define UARTA2_BUFFER_FULL              (UARTA2WriteIndex + 1) % UARTA2_BUF
95 */
96
97 /******
98 @函数名: EUSCIA2_IRQHandler
99 @入口参数: 无
100 @出口参数: 无
101 功能描述: 串口2数据接收
102 @作者: WUST
103 @日期: 2019年01月27日
104 *****/
105 void EUSCIA2_IRQHandler(void)
106 {
107     uint32_t status = MAP_UART_getEnabledInterruptStatus(EUSCI_A2_BASE);
108     if(status & EUSCI_A_UART_RECEIVE_INTERRUPT_FLAG)
109     {
110         uint8_t data=MAP_UART_receiveData(EUSCI_A2_BASE);
111         NCLink_Data_Prase_Prepare(data); //(UCA2RXBUF);
112     }
113     MAP_UART_clearInterruptFlag(EUSCI_A2_BASE, status);
114 }
115
116 /*
117     if(UARTA2_BUFFER_FULL)
118     {
119         //TODO: Buffer Overflow, add handler here
120     }
121     else
122     {
123         UARTA2Data[UARTA2WriteIndex] = data;
124         UARTA2_ADVANCE_WRITE_INDEX;
125     }
126 */
127 /******
128 //波特率计算公式:
129 // N=时钟主频/波特率
130 *****/
```

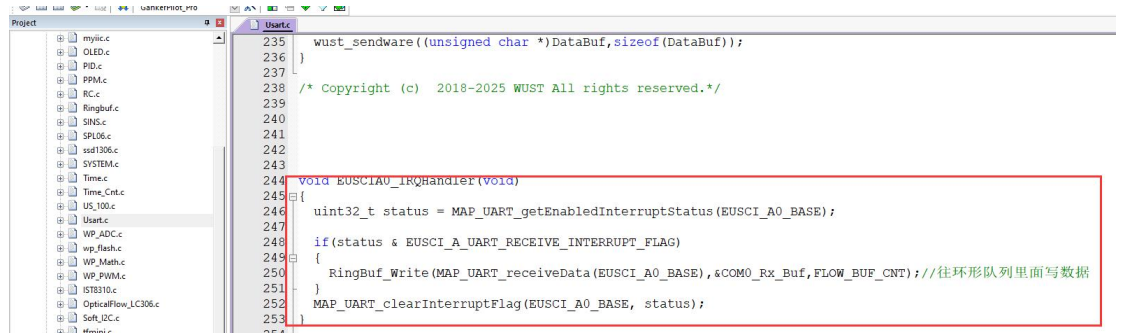
地面站数据解析

```
310 void EUSCIA1_IRQHandler(void)
311 {
312     uint32_t status = MAP_UART_getEnabledInterruptStatus(EUSCI_A1_BASE);
313
314     if(status & EUSCI_A_UART_RECEIVE_INTERRUPT_FLAG)
315     {
316         RingBuf_Write(MAP_UART_receiveData(EUSCI_A1_BASE), &COM1_Rx_Buf, SDK_Target_Length*2); //往环形队列里面写数据
317         if(COM1_Rx_Buf.Ring_Buff[0]!=0xFF)
318         {
319             COM1_Rx_Buf.Head=1;
320             COM1_Rx_Buf.Tail=0;
321         }
322     }
323     MAP_UART_clearInterruptFlag(EUSCI_A1_BASE, status);
324 }
325
```

OPENMV1 数据解析

```
01 void EUSCIA3_IRQHandler(void)
02 {
03     uint32_t status = MAP_UART_getEnabledInterruptStatus(EUSCI_A3_BASE);
04
05     if(status & EUSCI_A_UART_RECEIVE_INTERRUPT_FLAG)
06     {
07         RingBuf_Write(MAP_UART_receiveData(EUSCI_A3_BASE), &COM3_Rx_Buf, SDK_Target_Length*2); //往环形队列里面写数据
08         if(COM3_Rx_Buf.Ring_Buff[0]!=0xFF)
09         {
10             COM3_Rx_Buf.Head=1;
11             COM3_Rx_Buf.Tail=0;
12         }
13     }
14
15     MAP_UART_clearInterruptFlag(EUSCI_A3_BASE, status);
16 }
17
18
19
```

OPENMV2 数据解析



光流数据解析

综上所述，飞控程序运行时，存在三类主要的任务：

- 1、定时器任务调度中断函数；
- 2、其它外设中断函数（PPM 解析、串口中断等）；
- 3、While(1) 里面非实时函数；

要想程序运行稳定、串口数据尽可能地不丢帧，PPM 数据解析稳定，必须要考虑每个任务实际用时开销，对中断优先级进行合理的设计。

```
75  /*****
76  函数名: void NVIC_Configuration(void)
77  说明: 中断优先级定义
78  入口: 无
79  出口: 无
80  备注: 系统中断时间在整体设计时，已全部测试过最大开销，
81        确保程序运动的稳健性
82  注释者: WUST
83  *****/
84  void NVIC_Configuration(void)
85  {
86      sysTickUptime=0;
87      Interrupt_setPriorityGrouping(4);
88      MAP_Interrupt_setPriority(FAULT_SYSTICK,0x00);
89      MAP_Interrupt_setPriority(INT_T32_INT2,0x00);
90      MAP_Interrupt_setPriority(INT_EUSCIA2 ,0x20);//串口2 地面站 波特率921600
91      MAP_Interrupt_setPriority(INT_PORT6  ,0x20);//PPM US100
92      MAP_Interrupt_setPriority(INT_PORT3  ,0x40)//GY53L1X_IN2
93      MAP_Interrupt_setPriority(INT_PORT5  ,0x40)//GY53L1X_IN2
94
95      MAP_Interrupt_setPriority(INT_EUSCIA3 ,0x60)//串口3 OPENMV 波特率921600
96      MAP_Interrupt_setPriority(INT_EUSCIA0 ,0x80)//串口0 LC306 波特率19200
97      MAP_Interrupt_setPriority(INT_EUSCIB1 ,0xa0)//串口1 OPENMV 波特率921600
98
99      MAP_Interrupt_setPriority(INT_ADC14   ,0xc0);
100     MAP_Interrupt_setPriority(INT_TA2_0   ,0xe0);//任务调度器
101
102 }
```

在设计定时器任务调度中断函数时，需要对每一个子任务的时间开销进行统计，且定时器中断服务函数的调度周期要大于子任务的最大时间开销之和。在进行串口中断优先级设计时，需要把握以下几点：

- 1、多个串口通讯时，串口通讯波特率可以降低一点。
- 2、合计设计优先级，当存在不同波特率通讯时，通讯波特率高的串口中断优先级要高于波特率低的。
- 3、存在优先级高于串口中断的其它中断任务时，其它中断任务的总的最大时间开销也要考虑。