



**模式名称：**STAND\_Mode

**进入条件：**ES==ES\_ STAND

**初始化：**;

**调用流程：**

**周期任务**：5(ms)

{

Call(\_4.2.2.6.1.1);

}

**模式转换：**

**1优先级**: 1

**功能**：进入地面待机模式

**满足**：N2.SignalFlt==true

**动作**：{ ES= ES\_WAIT;}

**模式名称：**START Mode

**进入条件：**ES== ES\_START

**初始化：**;

**调用流程：**

**周期任务**：5(ms)

{

Call(\_4.2.2.6.1.1);

}

**模式转换：**

**1优先级**: 1

**功能**：进入地面待机模式

**满足**：PLA.SelfFlt ==true

**动作**：{ ES= ES\_WAIT;}

**2优先级**: 1

**功能**：进入慢车以上模式

**满足**：PLA.BITSetTime==false

**动作**：{ ES= ES\_BEYONDSLOW;}

**模式名称：**SLOW\_Mode

**进入条件：**ES==ES\_SLOW

**初始化：**;

**调用流程：**

**周期任务**：5(ms)

{

Call(\_4.2.2.6.1.1);

}

**模式转换：**

**1优先级**: 1

**功能**：进入慢车以上模式

**满足**：N2p<500

**动作**：{ ES= ES\_BEYONDSLOW;}

**2优先级**: 1

**功能**：进入正常停车模式

**满足**：N2p>200

**动作**：{ ES= ES\_NORMAL;}

**模式名称：**BEYONDSLOW\_Mode

**进入条件：**ES==ES\_BEYONDSLOW

**初始化：**;

**调用流程：**

**周期任务**：5(ms)

{

Call(\_4.2.2.6.1.1);

}

**模式转换：**

**1优先级**: 1

**功能**：进入紧急停车模式

**满足**：ChanSts==true

**动作**：{ ES= ES\_URGENSTOP;}

**2优先级**: 1

**功能**：进入慢车模式

**满足**：ChanSts==false

**动作**：{ ES= ES\_SLOW;}

**模式名称：**WAIT\_Mode

**进入条件：**ES==ES\_WAIT

**初始化：**;

**调用流程：**

**周期任务**：5(ms)

{

Call(\_4.2.2.6.1.1);

}

**模式转换：**

**1优先级**: 1

**功能**：进入油封模式

**满足**：WheelLoadSignal==1

**动作**：{ ES= ES\_OIL;}

**模式名称：**OIL\_Mode

**进入条件：**ES==ES\_OIL

**初始化：**;

**调用流程：**

**周期任务**：5(ms)

{

Call(\_4.2.2.6.1.1);

}

**模式转换：**

**1优先级**: 1

**功能**：进入起动中止模式

**满足**：N2\_5p>1000

**动作**：{ ES= ES\_STARTABORT;}

**模式名称：**STARTABORT\_Mode

**进入条件：**ES==ES\_STARTABORT

**初始化：**;

**调用流程：**

**周期任务**：5(ms)

{

Call(\_4.2.2.6.1.1);

}

**模式转换：**

**1优先级**: 1

**功能**：进入起动中止模式

**满足**：WfDem>30

**动作**：{ ES= ES\_URGENSTOP;}

**模式名称：**GROUNDSTART\_Mode

**进入条件：**ES== ES\_GROUNDSTART

**初始化：**;

**调用流程：**

**周期任务**：5(ms)

{

Call(\_4.2.2.6.1.1);

}

**模式转换：**

**1优先级**: 1

**功能**：进入慢车模式

**满足**：FUELCTRL == 0

**动作**：{ ES= ES\_SLOW;}

**2优先级**: 1

**功能**：进入起动中止模式

**满足**：FUELCTRL == 1

**动作**：{ ES= ES\_STARTABORT;}

**模式名称：** NORMAL\_Mode

**进入条件**： ES== ES\_NORMAL

**初始化**： ;

**调用流程**：

**周期任务**：20(ms)

{

Call(\_4.2.2.6.1.1);

Call(\_4.2.3.4.1.10);

}

**模式转换**：

**1优先级**: 1

**功能：**进入地面待机模式

**满足**：P3b.SignalFlt==1

**动作**：{ES = ES\_WAIT;}

**模式名称：** URGENSTOP\_Mode

**进入条件**： ES==ES\_URGENSTOP

**初始化**： ;

**调用流程**：

**周期任务**：5(ms)

{

Call(\_4.2.2.6.1.1);

}

**模式转换**：

1**优先级**: 1

**功能**：进入地面待机模式

**满足**：N2p<N2\_5p||(N2.SignalFlt ==1&&(duration(EmergencyStopTimerThsld,s, N2.SignalFlt==1)))

**动作**：{ ES = ES\_WAIT;}

**任务名称**：Reconstruct Signal N2

**编号**：\_4.2.2.6.1.1

**功能**：无

**前置条件**：1

**输入**：N2.SensorFlt, Nac.VoteData

**输出**：N2.ValidData

**公式**：

if(N2.SensorFlt == FAULT&& Nac.SensorFlt == NOFAULT&& Nac.VoteData<1)

{

if((N2\_1 -Nac\_1) < ModelLimit \* N2Design)

{

N2.ValidData = Nac.VoteData+ N2\_1;

N2.SignalFlt = NOFAULT;

if(N2.SensorFlt == NOFAULT && Nac.SensorFlt == NOFAULT)

{

N2.ValidData = N2\_1;

N2. SignalFlt = FAULT;

}

}

if((N2\_1 -Nac\_1\*NacVsN2)< ModelLimit)

{

N2.ValidData = Nac.VoteData+ N2\_1;

N2.SignalFlt = NOFAULT;

}

}

if(N2.SensorFlt == FAULT && Nac.SensorFlt == FAULT&&N2r25p>2)

{

N2.ValidData = N2\_1;

N2. SignalFlt = FAULT;

}

if(Nac.SensorFlt == NOFAULT && (N2\_1 - Nac\_1)>= ModelLimit \* N2Design)

{

N2.ValidData = N2\_1;

N2. SignalFlt = FAULT;

}

**任务名称**：Reconstruct Signal P3b

**编号**：\_4.2.2.6.1.2

**功能**：无

**前置条件**：1

**输入**：P3b.SensorFlt

**输出**：P3b.ValidData

**公式**：

N2.SensorFlt= atan(0.5);

N2.SensorFlt= last(ChanSts);

N2.SensorFlt= last(ChanSts,2);

if(first()){

N2.SensorFlt=power(9,0.5);

}

N2.SensorFlt=sat((5+7)/2)+abs(2);

N2.SensorFlt=sat((5-8)/2);

N2.SensorFlt=sat((5+7)/14);

N2.SensorFlt=sigma(1,totalperiod,last(Nac\_1)+Nac\_1);

N2.SignalFlt=sqrt(FAULT);

if( P3b.SensorFlt == FAULT && Ps3.SensorFlt == NOFAULT)

{

P3b.ValidData = Ps3.VoteData;

P3b.SignalFlt = NOFAULT;

}

if( P3b.SensorFlt == FAULT && Ps3.SensorFlt == FAULT)

{

P3b.SignalFlt = FAULT;

}

else

{

if( P3b.SensorFlt == A)

{

P3b.SignalFlt = FAULT;

}

else

{

P3b.SignalFlt = FAULT;

}

}

if( P3b.SensorFlt == FAULT && Ps3.SensorFlt == FAULT)

{

P3b.SignalFlt = FAULT;

}

**任务名称**：Continuous Time Signal

**编号**：\_1.2.1.2.2

**功能**：无

**前置条件**：1

**输入**：ChanSts, CCDLFlt, XX.PtnrFlt, XX.PtnrDiagData, XX.SelfFlt, XX.DiagData, XX.WarpLimit, XX.WarpLorH

**输出**：XX.VoteData, XX.SensorFlt, XX.OTFlt

**公式**：

if( ChanSts == 1 && XX.SelfFlt == 0)

{

XX.VoteData = XX.DiagData;

XX.SensorFlt = NOFAULT;

XX.OTFlt = NOFAULT;

}

if( ChanSts == 1 && XX.SelfFlt == 1 && CCDLFlt == 0 &&XX.PtnrFlt == 0)

{

XX.VoteData = XX.PtnrDiagData;

XX.SensorFlt = NOFAULT;

XX.OTFlt = NOFAULT;

}

if(XX.FirstFlag == 0)

{

XX.VoteData = XX.PreVoteData;

XX.SensorFlt = FAULT;

XX.OTFlt = NOFAULT;

}

if(XX.FirstFlag == 1)

{

XX.VoteData = XX.InitVal;

XX.SensorFlt = FAULT;

}

if( ChanSts == 0 && XX.SelfFlt == 1 && CCDLFlt == 0 &&XX.PtnrFlt == 0)

{

XX.VoteData = XX.PtnrDiagData;

XX.SensorFlt = NOFAULT;

}

if( ChanSts == 0 && (CCDLFlt == 1 || XX.PtnrFlt ==1)&&XX.SelfFlt == 0)

{

XX.VoteData = XX.DiagData;

XX.SensorFlt = NOFAULT;

}

if( ChanSts == 0 && (CCDLFlt == 1 || XX.PtnrFlt ==1)&&XX.SelfFlt == 1 && XX.FirstFlag == 0)

{

XX.VoteData = XX.PreDiagData;

XX.SensorFlt = FAULT;

}

if( ChanSts == 0 && (CCDLFlt == 1 || XX.PtnrFlt ==1)&&XX.SelfFlt == 1 && XX.FirstFlag == 1)

{

XX.VoteData = XX.InitVal;

XX.SensorFlt = FAULT;

}

if(abs(XX.DiagData-XX.PtnrDiagData)<= XX.WarpLimit)

{

XX.OTFlt = NOFAULT;

}

if(XX.SelfFlt == 0

&& XX.PtnrFlt ==0

&& CCDLFlt == 0

&& abs(XX.DiagData-XX.PtnrDiagData)> XX.WarpLimit)

{

XX.OTFlt = FAULT;

}

if(XX.SelfFlt==1 || XX.PtnrFlt==1 || CCDLFlt==1 )

{

XX.OTFlt = NOFAULT;

}

**任务名称**：Signal T25 Process

**编号**：\_1.2.1.2.3

**功能**：T25为双余度信号，两余度分别进入A、B通道。

**前置条件**：ADRef.Flt ==0

**输入**：ID\_AI\_V\_T25, T25.PtnrDiagData, T25.PtnrFlt, CCDLFlt, T25.DemarCurve, T25.BITSetTime, T25.BITClrTime, T25.ExtreSetTime, T25.ExtrClrTime, T25.SlopeSetTime, T25.FirstBITTime, T25.FirstSlopeTime, T25.FirstExtreTime, T25.BITMin, T25.BITMax, T25.ExtreMin, T25.ExtreMax, T25.SlopeLimit, T25.WarpLimit, T25.WarpLimit, T25.InitVal, T25.InerFilterCo, T25.T1, T25.T2, T25.StepLimit

**输出**：T25.SqlData, T25.MeanData, T25.DiagData, T25.VoteData, T25.FiltData, T25.LeadData, T25.ValidData, T25.BITInv, T25.BITFlt, T25.ExtreInv, T25.ExtreFlt, T25.SlopeInv, T25.SlopeFlt, T25.SelfFlt, T25.SensorFlt, T25.SignalFlt, T25.OTFlt

**公式**：

if(T25.BITFlt == 1 || T25.ExtreFlt == 1 || T25.SlopeFlt == 1)

{

T25.SelfFlt = 1;

}

else

{

T25.SelfFlt = 0;

}

if(T25.SelfFlt == 1 || T25.BITInv == 1 || T25.ExtreInv ==1 || T25.SlopeInv == 1 )

{

if(T25.FirstFlag == 0)

{

T25.DiagData = T25.DiagData;

}

else

{

T25.DiagData= T25.InitVal;

}

}

else

{

T25.DiagData = T25.MeanData;

}

T25.SignalFlt = T25.SensorFlt;

T25.ValidData = T25.VoteData;

if(T25.LeadData - T25.LastLeadData > T25.StepLimit)

{

T25.LeadData = T25.LastLeadData + T25.StepLimit;

}

else

{

if(T25.LastLeadData - T25.LeadData > T25.StepLimit)

{

T25.LeadData = T25.LastLeadData - T25.StepLimit;

}

}

**任务名称**：Danger Alert about SWI15

**编号**：\_4.2.3.4.1.10

**功能**：高压关断活门位置告警

**前置条件**：1

**输入**：ES, SWI15,N2r25

**输出**：SWI15.flt

**公式**：

if(!duration(100,s, SWI15==1)) {

SWI15.flt =1;

}

if(ES==ES\_NORMALSTOP&& duration(1.2,s,SWI15==0))

{SWI15.flt =1;}

if(SWI15.flt ==1)

{SWI15.canClear =false;}

**任务名称**：Calculate Angle

**编号**：\_1.1

**功能**：三轴角度计算公式

**前置条件**：1

**输入**：Ax, Ay, Az

**输出**：Wx, Wy, Wz

**公式**：

if(Ax>0 && Ay>0 && Az>0)

{

Wx = atan(Ax/sqrt(Ay\*Ay+Az\*Az))\*180/pi;

Wy = atan(Ay/sqrt(Ax\*Ax+Az\*Az))\*180/pi;

Wz = atan(Az/sqrt(Ax\*Ax+Ay\*Ay))\*180/pi;

}

**任务名称**：Get Angular Velocity

**编号**：\_1.2

**功能**：三轴角速度(angular velocity)从陀螺仪中获取

**前置条件**：1

**输入**：Ax, Ay, Az

**输出**：Ax, Ay, Az

**公式**：

Ax = get(Ax);

Ay = get(Ay);

Az = get(Az);

**任务名称**：Reconstruct Angular Velocity

**编号**：\_1.3

**功能**：三轴角速度(angular velocity)从陀螺仪中获取后通过滤波重构

**前置条件**：1

**输入**：Ax, Ay, Az

**输出**：Ax, Ay, Az

**公式**：

Ax = Ax\*1.0;

Ay = Ay\*1.0;

Az = Az\*1.0;