

# 载人航天总体部自动判读语言 V2.0 语法

## 一. 简介

载人航天总体部目前使用的自动判读语法为 V2.0 版本，在原有基础（以下称为 V1.0）上扩展了状态机描述、延迟时间等内容，具有判据表述能力强、延迟判读知识可以灵活书写等特点。

本文对自动判读语言 V2.0 语法进行基本描述，并给出具体实例。

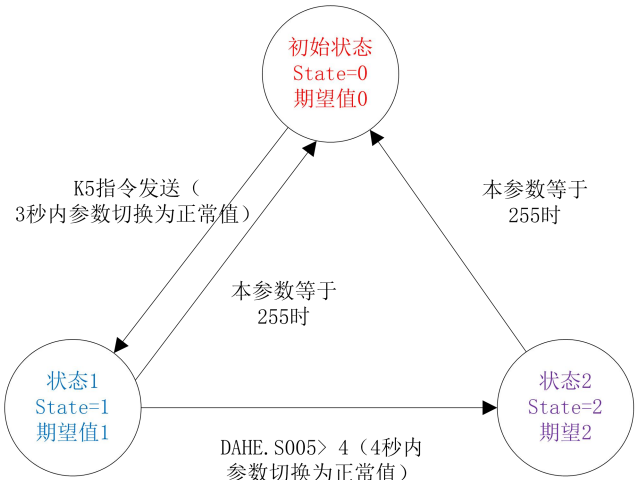
## 二. 整体结构

载人航天总体部自动判读 V2.0 语法在状态机基础上构建，参考了常用的脚本语言描述方式。在判读过程中，判读人员实际是先确定参数所处的状态，再根据状态对参数进行判读，延迟等因素也与参数状态相关。

相对应地，自动判读 V2.0 的参数判据由初始化域（变量执行初始化）、状态变换域和判读结论域三部分组成，初始化域只包含局部变量的定义和全局变量赋值；状态变换域只包含状态和变量的赋值，承担延迟判读相关信息的书写；判读结论域只包含某状态下的判读知识，不再接受延迟时间信息。

自动判读 V2.0 语法的最佳实践，就是先写状态切换，后根据状态写值域范围。其中指令导致的状态切换时，用了新的描述 COMD.\*\*、SETP.\*\*等，表示指令的瞬时发送，而不是用原有的 gzCOMD\*\*，gzSETP\*\*的持续量，因为使用持续量无法良好的描述指令发送的顺序、次数等信息。

考虑如下一个参数的状态变换图：



完整示例如下：

### //语法举例

**InitRegion** //标识初始化域开始

this.state = 0; //将状态初始化为 0

local a=0; //local 表示局部变量。不用 local 标识的变量，均当全局变量处理

**EndInitRegion** //标识初始化域结束

删除[任亮]: 1

**StateRegion** //标识状态变化域开始

if((this.state ==0)&&(COMD.K5 (Filter 4s))) then this.state = 1; //若参数当前状态为 0，发送了 K5 指令后，则参数进入状态 1，且在 4 秒的时间内切换为新状态的正确值

else if((this.state ==1)&&(DAHE.S005> 4 (Filter 5s))) then this.state = 2; //若参数当前状态为 1，当参数 DAHE.S005 变为大于 4 的值后，则参数进入状态 2，且在 5 秒的时间内切换为新状态的正确值

elseif(this.value==255)then this.state=0; //若参数的遥测值变为 255，则进入状态 0  
end

**EndStateRegion** //标识状态变化域结束

**ResultRegion** //标识判读域开始

if(this.state ==0) then EV=5, ERR=2; //期望值在 3-7 范围内为正确，否则为异常

elseif(this.state ==1) then

a=a+1;

if((DAHE.S006 >= 0)&&(a>5)) then RET=Unusual;end; //为前一个 elseif then 语句的嵌套语句。可以有多个嵌套，注意嵌套语句以 end 结尾。Unusual 相当于原来的 False，详见说明下一节。

elseif(this.state ==2) then RET=Normal; //Normal 相当于原来的 True,详见下一节  
end

**EndResultRegion** //标识判读域结束

## 三. 细节描述

自动判读 V2.0 语法详细说明如下：

1. 完整判读由三部分组成：初始化域（InitRegion 标示开始，EndInitRegion 标示结束）、状态变换域（StateRegion 标示开始，EndStateRegion 标示结束）和判读结论域（ResultRegion 标示开始，EndResultRegion 标示结束）。

初始化域是局部变量的定义和全局变量的赋值；状态变换域是对状态和全局变量的赋值，承担延迟判读相关信息的获取；判读结论域是对某状态下的判读知识，且不再接受延迟时间信息。

无论是状态变化域还是判读结论域，其都可以利用定义的局部变量和全

局变量辅助逻辑执行，如上述例子中的“local a=0;”；也都可以定义较为复杂的逻辑。逻辑最大限度保持了与 V1.0 版本的一致，其形式如下：

**if 条件 then 执行语句;elseif 条件 then 执行语句;end;**

- 2. 示例中的 local 为新增关键字，表示局部变量。不用 local 标识的变量，均当做其他参数变量、全局变量或局部变量来处理；
- 3. 示例中的 value 和 state 是参数的两个基本属性，value 表示该参数的值（电压、电流、状态量等实测遥测参数值）；state 是该参数的状态标识，可以为大于等于 0 的任意 32 位整型数字；
- 4. 示例中的 this 保留字指参数本身，如 DATE.S001 参数的规则中，this.value 等价与 DATE.S001.value，其中“DATE.S001.value”中的“.value”可以省略不写，但“this.value”中的“.value”不可省略；
- 5. 示例中的 if 语句是整个逻辑的核心，有三种形式：

```
If.....then.....
end;

If.....then.....
else.....
end;

if.....then.....
elseif.....then.....
..--->可多个elseif.....then.....
end;
```

每个完整逻辑均由 if 开头以 end 结尾的结束以“;”为标志。逻辑可以嵌套，如下所示：

**if 条件 then**  
    **if 条件 then 执行语句;elseif 条件 then 执行语句;end;**  
**elseif 条件 then 执行语句;**  
**else 执行语句;**  
**end;**

- 6. 示例中的 Filter 为新增的关键字。Filter 后紧跟一个数字为延迟判读的的时间，如“if (COMD.K5 (Filter 4s)) then this.state = 2”标示 K5 指令发送

后进入状态 2(状态 2 下 4 秒以内的报错忽略),“if (DAHE.S005> 4 (Filter 5)) then this.state = 1” 标示 S005 参数大于 4 后进入状态 1 (状态 1 下 5 秒以内的报错忽略)。通过以上方式替代了原 V1.0 延迟判读语法。

7. 参数用“数据流.参数序号”的方式进行标示,如 DATE.S001 表示 DATE 数据流的 S001 参数,其中 DATE 是数据流类型,表示工程遥测参数,S001 是参数序号,参数序号与参数代号的对应关系详见《自动判读规则中参数序号与参数代号对照表》;
8. 指令发送事件用“通道名.指令号”的方式进行标示,如 COMD.K1 标示 COMD 通道的 K1 指令发送;
9. 每个指令对应一个全局变量,用“gz+通道名+指令号”的方式进行标示,如 gzCOMDK1 表示 K1 的发送状态量;
10. 支持 RET (返回值)、EU (参数上限)、EL (参数下限)、EV (参数期望值)、ERR (参数期望值偏差范围)等 V1.0 版语法。

在自动判读 V2.0 中,还有以下特性可以利用:

1. 支持全局变量定义和使用,全局变量依然为 gzGLOGBAL 打头,由系统管理员添加。全局变量可以在判读开始前由系统管理员进行设置,从而实现比如设备未安装等需求(相关参数可以利用全局变量进行判读书写,安装和拆装后要求系统管理员修改全局变量的值即可,无需修改判据);
2. 支持局部变量的定义和使用,如在 InitRegion 中加入 local a;就增加了一个叫做 a 的局部变量,可以在判据中进行赋值和使用,此值的使用不会干扰其他的参数;
3. 增加了对十六进制数的支持,如 255 可以表示为 0XFF;
4. 增加了对参数触发延迟判读的支持,如 `if((this.state==1) && ((DAHE.S005>5)(Filter 2.5s))) then this.state=2;`表示若 DAHE.S005>5,则进入状态 2,且状态 2 允许有 2.5 秒的参数切换时间;
5. 增加了对按位操作&, |, ……的支持;
6. 新增了 pre 关键字,用来标示参数之前的值,如 DATE.S001.pre1 表示

- DATE.S001 上一帧的值;
7. 为了进行分级报错, 新增了 Normal、Unusual、Error、SeriousError 关键字, 分别表示正确、异常、错误、严重错误 4 种不同级别的判断结论, 取代了原来的 True 和 False。在用户显示界面上上述 4 种结论均以不同的颜色区分显示。
  8. 新增了部分运算符, 目前可支持+ - \* / ^ (加减乘除幂)、< > <= >= == ~=、&& || !、&、|、~、<<、>>;
  9. 可以由 OSTIME 获取系统当前时间(为一个整数, 表示秒数), 如 local a; a=OSTIME; 即将系统当前时间暂存到局部变量 a 中。

此外, V2.0 语法将部分支持 1.0 版本语法, 但 **V1.0 语法书直接书写将不具备延迟判读能力**。

#### 四. 移植示例

以下表格是若干具体由 V1.0 语法向 V2.0 语法转换的实例, 分别用 V1.0 语法和 V2.0 语法描述, 仅供参考。

示例类型	老版（V1.0）判据	新版（V2.0）判据
基本表达	EL=95,EU=105;	ResultRegion EL=95,EU=105; EndResultRegion
简单状态转换	if (gzSETPPSS025==1) then   EL=0,EU=1; if (gzSETPPSS024==1) then   EL=3.5,EU=5; EL=3.5,EU=5;	InitRegion this.state=1; EndInitRegion  StateRegion if((this.state==1) && (SETP.PSS025(Filter 2.5s))) then this.state=2; elseif ((this.state==2) && (SETP.PSS024(Filter 2.5s))) then this.state=1; end EndStateRegion  ResultRegion if(this.state==1) then EL=3.5,EU=5; else EL=0,EU=1; end EndResultRegion
较为复杂的状态转换	EL=-60,EU=-40; if ((gzCOMDP347==1)  (gzCOMDK495==1)) then   EL=-60,EU=3; if ((gzCOMDP347==1)  (gzCOMDK495==1))&&((gzCOMDP343==1)  (gzCOMDK491==1)  (gzCOMDP349==1)  (gzCOMDK497==1)) then   EL=24,EU=37; if	InitRegion this.state=1; EndInitRegion  StateRegion if((this.state==1)&&(COMD.P347(Filter 4s)  COMD.K495(Filter 4s))) then this.state=2;

	((gzCOMDP347==1)  ((gzCOMDK495==1))&&((gzCOMDP344==1)  ((gzCOMDK492==1)  ((gzCOMDP350==1)  ((gzCOMDK498==1)) then EL=0,EU=3;	elseif((this.state==2    this.state==4)&&(COMD.P343(Filter 4s)  COMD.K491(Filter 4s))) then this.state=3; elseif((this.state==2    this.state==3)&&(COMD.P344(Filter 4s)  COMD.K492(Filter 4s))) then this.state=4; elseif((this.state!=1)&&(COMD.P348(Filter 4s)  COMD.K496(Filter 4s))) then this.state=1; end EndStateRegion  ResultRegion if(this.state==1) then EL=-60,EU=-40; elseif(this.state==2) then EL=-60,EU=3; elseif(this.state==3) then EL=24,EU=37; else EL=0,EU=3; end EndResultRegion
十六进制表示	RET=1; if (DAHE.I152==60304)&&((DAHE.I162&(3*64*256))!=0) then RET=0; if (DAHE.I152==60304)&&((DAHE.I162&3)!=0) then RET=0;	ResultRegion RET=Normal; if (DAHE.I152==0xEB90)&&((DAHE.I162&0xC000)!=0) then RET=Unusual; elseif (DAHE.I152==0xEB90)&&((DAHE.I162&0x3)!=0) then RET=Unusual; end EndResultRegion
历史值	if (DATE.A010 - DATE.A010.L9 == 4) then RET=1; if (DATE.A010 - DATE.A010.L9 != 4) then RET=0;	InitRegion this.state=1; EndInitRegion  StateRegion

		<pre>if(this.value - this.pre9 == 4) then this.state=2; elseif(this.value - this.pre9 != 4) then this.state=3; end EndStateRegion  ResultRegion if(this.state==2) then RET=Normal; elseif(this.state==3) then RET=Unusual; end EndResultRegion</pre>
--	--	--

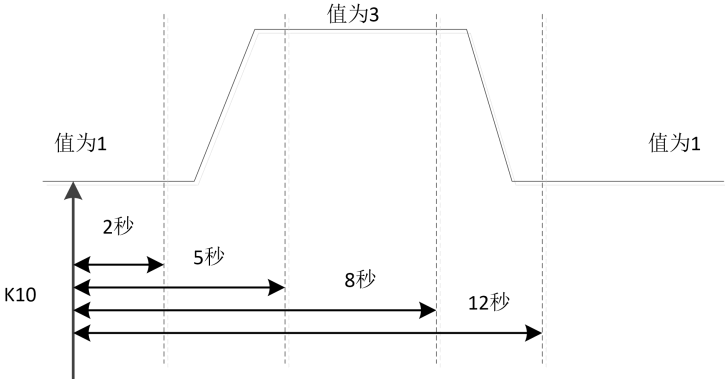
五. V2.0 若干最佳写法

本节从若干典型参数入手，介绍 V2.0 的若干最佳写法。

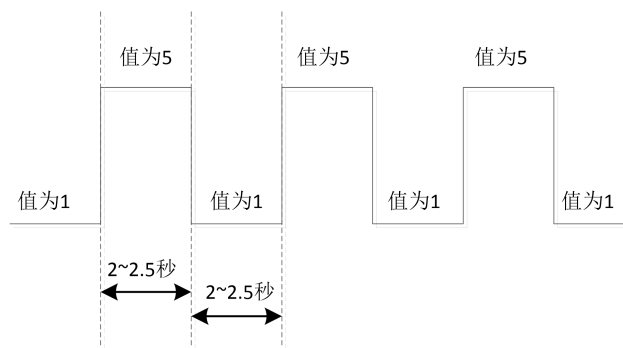
序号	状态图	V2.0 判据	备注
----	-----	---------	----



1	<pre>graph TD     S1((主机模式, 设备关闭)) -- PSS001 --&gt; S2((主机模式, 设备开启))     S2 -- PSS002 --&gt; S1     S2 -- "K1或P1 K2或P2" --&gt; S3((备机模式, 设备开启))     S3 -- PSS002 --&gt; S4((备机模式, 设备关闭))     S4 -- PSS001 --&gt; S3     S3 -- "K1或P1 K2或P2" --&gt; S2     S1 -.-&gt; V1[值域: -1~1]     S4 -.-&gt; V1     S2 -.-&gt; V2[值域: 3~5]     S3 -.-&gt; V2</pre>	<p><b>InitRegion</b></p> <pre>this.state=1; EndInitRegion</pre> <p><b>StateRegion</b><pre>if((this.state==1)&amp;&amp;(SETP.PSS001(Filter 4s))) then this.state=2; elseif((this.state==2 )&amp;&amp;(SETP.PSS002(Filter 4s))) then this.state=1; elseif((this.state==2)&amp;&amp;(COMD.K1(Filter 4s)  COMD.P1(Filter 4s))) then this.state=3; elseif((this.state==3)&amp;&amp;(COMD.K2(Filter 4s)  COMD.P2(Filter 4s))) then this.state=2; elseif((this.state==3)&amp;&amp;(SETP.PSS002(Filter 4s))) then this.state=4; elseif((this.state==4 )&amp;&amp;(SETP.PSS001(Filter 4s))) then this.state=3; end EndStateRegion</pre><p><b>ResultRegion</b><pre>if(this.state==1) then EL=-1,EU=1;           //主、关 elseif(this.state==2) then EL=3,EU=5;       //主，开 elseif(this.state==3) then EL=-1,EU=1;      //备，开 else EL=-1,EU=1;                            //备，关 end EndResultRegion</pre></p></p>	虽然只有 2 个值域，但是蕴含的意义是 4 各状态
---	--	---	---------------------------

2		<pre>InitRegion local K10Time=-1; local K10DiffTime=-1; this.state=1; EndInitRegion  StateRegion if(COMD.K10) then K10Time=OSTIME;end K10DiffTime = OSTIME-K10Time; if((K10Time!=-1)&amp;&amp;(K10DiffTime&gt;2)&amp;&amp;(K10DiffTime&lt;5)) then this.state=2; elseif((K10Time!=-1)&amp;&amp;(K10DiffTime&gt;5)&amp;&amp;(K10DiffTime&lt;8)) then this.state=3; elseif((K10Time!=-1)&amp;&amp;(K10DiffTime&gt;8)&amp;&amp;(K10DiffTime&lt;12)) then this.state=4; elseif((K10Time!=-1)&amp;&amp;(K10DiffTime&gt;12)) then this.state=1; end EndStateRegion  ResultRegion if(this.state==1) then EV=1,ERR=0; elseif(this.state==2) then EL=1,EU=3; elseif(this.state==3) then EV=3,ERR=0; elseif(this.state==4) then EL=1,EU=3; end EndResultRegion</pre>	<p>通过 <b>OSTIME</b> 和自定义的 <b>local</b> 局部变量，可以书写和时间差相关的判据</p>
---	--	--	---

3



InitRegion

```
this.state=0;
```

```
local state1StartTime = -1;
```

```
local state2StartTime = -1;
```

EndInitRegion

StateRegion

```
if(this.value == 1 && this.state!=1)
```

```
then    this.state=1;
```

```
state1StartTime=OSTIME ;
```

```
elseif(this.value == 5 && this.state!=2)
```

```
then    this.state=2;
```

```
state2StartTime=OSTIME;
```

end

EndStateRegion

## ResultRegion

```
if(this.state==1) then
```

```
if(this.value!=1) then RET=Unusual;
```

```
elseif(state2StartTime !=-1 && ((state1StartTime-state2StartTime
2))|(state1StartTime-state2StartTime>2.5))) then RET=Unusual;
```

```

else RET=Normal;

```

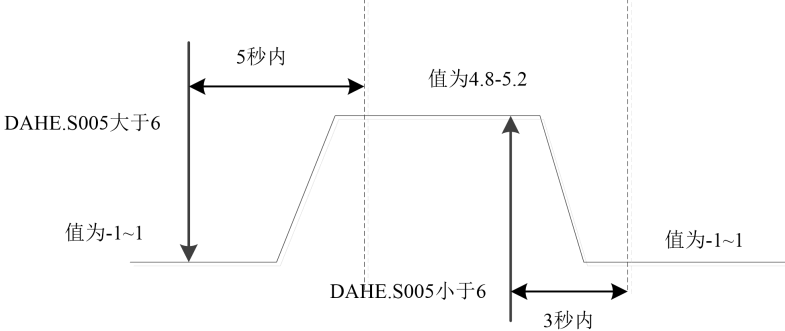
end

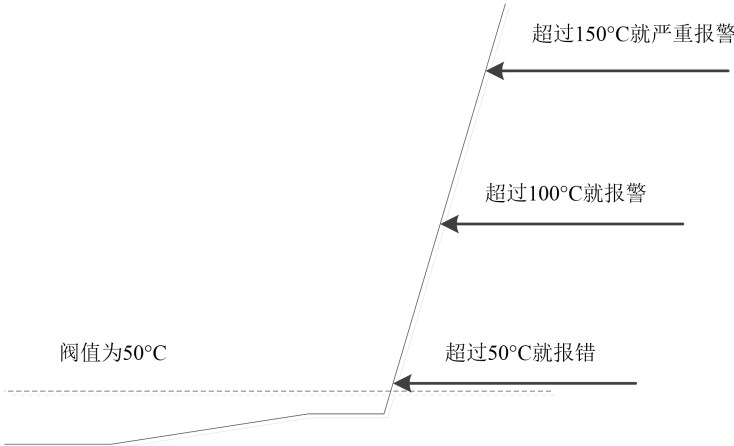
```
elseif(this.state==2) then
```

```
if(this.value!=5) then RET=Unusual;
```

```
elseif(state1StartTime !=-1 && ((state2StartTime-state1StartTime
```

通过 **OSTIME** 和自定义的 **local** 局部变量, 可以对方波信号进行判读

		<pre>&lt;2)  ((state2StartTime-state1StartTime &gt;2.5))) then RET=Unusual;     else RET=Normal;   end end EndResultRegion</pre>	
4		<pre>InitRegion this.state=1; EndInitRegion  StateRegion if((this.state==1)&amp;&amp;((DAHE.S005&gt;6)(Filter 5s))) then this.state=2; elseif((this.state==2)&amp;&amp;((DAHE.S005&lt;3)(Filter 3s))) then this.state=1; end EndStateRegion  ResultRegion if(this.state==1) then EL=-1,EU=1; elseif(this.state==2) then EV=5,ERR=0.2; end EndResultRegion</pre>	可实现由参数判 参数的延迟判读

5	 <p>阈值为50°C</p> <p>超过50°C就报错</p> <p>超过100°C就报警</p> <p>超过150°C就严重报警</p>	<pre>ResultRegion if(this.value&lt;=50) then RET=Normal; elseif(this.value&gt;50 &amp;&amp; this.value&lt;=100) then RET=Unusual; elseif(this.value&gt;100 &amp;&amp; this.value&lt;=150) then RET=Error; elseif(this.value&gt;150) then RET=ServerError; end EndResultRegion</pre>	可实现分级报错
---	---	---	---------