

# 情景

## 一、关键字

### 1.1、state: 启动标识

- a、 idle 表示设备启动时不执行，除非收到相应的 run 命令。
- b、 run 表示设备启动时执行情景。
- c、 stop 表示情景在任务情景下都不执行。
- d、 pause 表示情景处于暂停状态，设备启动是加载情景到内存中并让其处于暂停状态。
- e、 当没有 state 状态标识的情景时，认为其 state 状态等同于“run”。

例：

```
{
  state: "run",
  "send": "addr:010001010000000C,keyID:02 ,port:01,cmd:02,option:01,aID:009020,dlen:1,data:01"
}
```

### 1.2、if: 条件语句

if 条件语句，判断其内部的 reason 语句。条件为真，进入语句块并执行内部的操作，否则判断紧跟其后的 else 语句是否存在,存在则运行 else 语句块的内容，如果不存在 else 语句块则继续往下运行。

例：if 里的条件为真执行 i++，为假执行 k++;

```
{
  state: "run"
  "if": {
    "reason": "time(z=8,time=21:2:01)",
    "do": "i++",
  }
  "else": {"do": "k++"}
}
```

### 1.3、while: 循环语句

while 语句判断其内部的 reason 语句。条件为真，则进入语句块并执行内部的操作。否则执行下一条指令。当其内部的 reason 值为数字时，非 0 为真。使用方式：

例 1：死循环一直执行 i++

```
{
  "while": {
    "reason": "1",
    "do": "i++"
  }
}
```

例 2: 执行 10 次 i++

```
{ "while": {  
    "reason": "i<10",  
    "do": "i++"  
  }  
}
```

## 1.4、delay: 延迟

延迟的毫秒数，误差正负 50 毫秒。

例：“delay”:1000”。情景将暂停运行，并等待 1000 毫秒后再次执行下一条语句。例中执行完 10 次 i++将需要 10 秒钟时间。

```
{ "while": {  
    "reason": "i<10",  
    "delay": 1000,  
    "do": "i++"  
  }  
}
```

## 1.5、send: 发送消息

把消息发送给目标设备，也可以发送给自己，参数有：

a、addr:0000000000000001。(必须参数)

b、keyID:00。(必须参数)

c、key:xxxxxxxxxxxxxxxxxx。

d、port:xx。(必须参数)

e、aID:xxxx。(必须参数)

f、option:xx。(必须参数)

g、cmd:xx。(必须参数)

h、dlen:xx。(必须与 data 成对出现)

i、data:xx。(data 的值可以是 16 进字符串也可以是关键字 wait.data)

注：当使用 wait.data 关键字将忽略 dlen，以接收到的设备数据长度为准。同时使用 test 命令对此条 命令无效。

例 1:

```
{  
  state: "run",  
  "send": "addr:010001010000000C,keyID:02,port:01,cmd:02,option:01,aID:009020,dlen:1,data:01"  
}
```

例 2:

```
{
  state:"run",
  "if"{
    "reason": "wait(addr=0100010100000019,point=04,cmd=08,option=01,aID=008010),
    "send": "addr:010001010000000C,keyID:02 ,port:01,cmd:02,option:01,aID:009020,dlen:wait.dlen,data:wait.data"
  }
}
```

## 1.6、read: 读设备消息

a、重复比较设备消息，直到条件为真或超时

b、当发生超时后，如果后面有紧跟的 exception 语句，将执行 exception 语句内的内容。

例：先用 send 发送读设备的命令，再用 read 不断读取消息。主要为解决读设备信息，又不能立刻接着往下执行指令的情况。

```
{
  "send": "addr:010001010000000C,keyID:02 ,port:01,cmd:02,option:01,aID:009020,dlen:01,data:00 "
  "read": {
    "reason": "wait(addr=0100010100000019,point=04,cmd=08,option=01,aID=008010) || expire:5000",
    "do": "i++"
  },
  "exception" {
    "do": "k++"
  }
  s++;
}
```

## 1.7、do: 执行语句

a、执行加法减法运算,i++, i--或 i=i+1,j=j+10,j=i-j;等。例: "do": "i=i+10"

b、执行乘法除法运算, i=a\*b 或 i=j/10。例"do": "i=10\*i"。

c、执行赋值。如 i=0、i=100。

例：执行 10 次 i++;

```
{
  "do": "i=0",
  "while": {
    "reason": "i<10",
    "do": "i++"
  }
}
```

## 1.8、join：加入某情景之后执行

a、暂停当前情景，直到某情景执行完成之后再执行执行，例：

“join”：“testA.vm”。暂停执行当前情景，直到 testA.vm 执行完成再继续接着执行。

例：testB.vm 在等待 testA 执行完之后才能执行。

testA.vm:

```
{
  "do": "i=0",
  "while": {
    "reason": "i<10",
    "do": "i++"
  }
}
```

testB.vm

```
{
  "join": "testA.vm",
  "do": "i=0",
  "while": {
    "reason": "1",
    "do": "i++"
  }
}
```

## 1.9、active：激活情景

a、激活处于暂停状态下的情景，例：

“active”：“testA.vm”。激活情景 testA.vm，当 testA.vm 处于暂停状态，接收到此命令将得以继续运行。

例：当 testA.vm 执行完后，testB 将从暂停状态切换到运行状态。

testA.vm

```
{
  "do": "i=0",
  "while": {
    "reason": "i<10",
    "do": "i++"
  }
  "active": "testB.vm",
}
```

testB.vm 处于暂停状态

```
{
  "state": "pause",
  "do": "i=0",
  "while": {
    "reason": "1",
    "do": "i++"
  }
}
```

## 二、 支持的方法

### 2.1、time 时间比较

- a、与本地系统时间比较，系统时间与比较的时间相等返回 0，系统时间小于比较的时间返回-1，系统时间大于比较的时间返回 1。
- b、参数可选，没有提供的参数将不参与比较：
  - z:表示时区。如果不设置时区将用系统时区代替。
  - date:表示日期。格式必须是 yy-mm-dd。
  - time:表示时间。格式必须是 hh:mm:ss。
  - wd:表示星期几。格式是整数。

例：time(z=8,date=2018-10-01,time=21:02:01)，time(z=8,wd=3,time=21:02:01)

### 2.1、wait 消息并比较

- a、等待设备的消息并比较消息，为真返回 1，为假返回 0。
- b、参数可选，没有提供的参数将不参与比较：

addr:匹配地址。  
port:匹配端口号。  
keyID:匹配权限。  
aID:匹配 aID。  
option:匹配 option。  
cmd:匹配 cmd。  
dlen:匹配数据长度。  
data:匹配数据。数据支持大于，小于，等于匹配。

例：wait(addr=0100010100000019,point=04,cmd=08,option=01,aID=008010)或：

wait(addr=0100010100000019,point=04,cmd=08,option =01,aID=008010, data>100)

## 三、 例子

### 3.1、设备启动时只运行 1 次的情景：

```
{
  "state": "run"
  "send": "addr:010001010000000C,keyID:02 port:01,option:02,cmd:01,aID:009020,dlen:1,data:01",
  "send": "addr:0100010100000012,keyID:02, port:04, option:02,cmd:01,aID:009020,dlen:1,data:01",
  "send": "addr:0100010100000012,keyID:02 port:03, option:02,cmd:01,aID:009020,dlen:1,data:01",
  "send": "addr:0100010100000012,keyID:02, port:02, option:02,cmd:01,aID:009020,dlen:1,data:01",
  "send": "addr:0100010100000012,keyID:02, port:01, option:02,cmd:01,aID:009020,dlen:1,data:01"
}
```

### 3.2、定时情景：

```
{
  "while": {
    "reason": "1",
    "if": {
      "reason": "time(z=8,time=21:2:01)",
      "send": "addr:010001010000000C,keyID:02 port:01,option:02,cmd:01,aID:009020,dlen:1,data:01",
      "send": "addr:0100010100000012,keyID:02, port:04, option:02,cmd:01,aID:009020,dlen:1,data:01",
      "send": "addr:0100010100000012,keyID:02 port:03, option:02,cmd:01,aID:009020,dlen:1,data:01",
      "send": "addr:0100010100000012,keyID:02, port:02, option:02,cmd:01,aID:009020,dlen:1,data:01",
      "send": "addr:0100010100000012,keyID:02, port:01, option:02,cmd:01,aID:009020,dlen:1,data:01"
    }
  }
}
```

### 3.3、区间定时：

```
{
  "while": {
    "reason": "1",
    "if": {
      "reason": "time(z=8,time=21:2:01)==1&& time(z=8,time=21:5:01)==-1",
      "send": "addr:010001010000000C,keyID:02 port:01,option:02,cmd:01,aID:009020,dlen:1,data:01",
      "send": "addr:0100010100000012,keyID:02, port:04, option:02,cmd:01,aID:009020,dlen:1,data:01",
      "send": "addr:0100010100000012,keyID:02 port:03, option:02,cmd:01,aID:009020,dlen:1,data:01",
      "send": "addr:0100010100000012,keyID:02, port:02, option:02,cmd:01,aID:009020,dlen:1,data:01",
      "send": "addr:0100010100000012,keyID:02, port:01, option:02,cmd:01,aID:009020,dlen:1,data:01"
    }
  }
}
```

### 3.4、读传感器控制设备：

```
{
  "while": {
    "reason": "1",
    "send": "addr:010001010000000C,keyID:02 port:01,option:00,cmd:02,aID:009020,dlen:1,data:01",
    "read": {
      "reason": "wait(addr=010001010000000C,point=04,option=08,cmd=01,aID=008010,data>200)|| expire:5000",
      "send": "addr:010001010000000C,keyID:02 port:01,option:02,cmd:01,aID:009020,dlen:1,data:01",
      "send": "addr:0100010100000012,keyID:02, port:04, option:02,cmd:01,aID:009020,dlen:1,data:01",
      "send": "addr:0100010100000012,keyID:02 port:03, option:02,cmd:01,aID:009020,dlen:1,data:01",
      "send": "addr:0100010100000012,keyID:02, port:02, option:02,cmd:01,aID:009020,dlen:1,data:01",
      "send": "addr:0100010100000012,keyID:02, port:01, option:02,cmd:01,aID:009020,dlen:1,data:01"
    }
  }
}
```