Interface Manipulation Language （IML）设计概要

接口操作语言（Interface Manipulation Language ，简称IML）是专用于嵌入式设备接口测试的编程语言；其设计目的是为简化测试用例编写，提升测试效率；IML通过类似自然语言的接口控制语句实现高效的测试用例开发，采用易于理解的线性编程方式设计可异步执行的用例，非常易于掌握，且具有很高的执行效率。

要点：

保留字全部小写、动词开头、分号结束

用例入口函数 function run

全局变量只能在主函数入口处声明

1. 数据类型

**int**

**bool**

**real**

**string**

**buffer**

**[ ]**

1. 注释格式

#至行尾的内容位注释

#region #endregion 之间的内容代表一个可折叠的代码块

1. 控制语句

变量声明：

**declear** *protocol @varname；*

**declare** *int @i = 1;*

**declare** *int @var1, string @var2*

**declear** type @@globarvarname

变量赋值：

**set** *@varname （..=default, ..=<..>, .seg1=1, .seg2=<colname>）;*

**set** *@i=2;*

收发单帧数据：

**send** @varname **on** channelname **timeout** 20;

**recve** @varname **on** channelname **timeout** 20;

启动周期收发：

**start send** outinterfacename **on** channelname **interval** 10 **repeated** 99 **continued** 1000 **until** cond\_exp;

**start recve** ininterfacename **on** channelname **interval** 10 **repeated** 99 **continued** 1000 **until** cond\_exp;

停止周期收发：

**stop send** **on** channelname;

**stop recve** **on** channelname;

结果验证：

**check** @varname>10, @var2<91,

**check** @protoname (.., .seg1/.@var1, .seg2>9 && seg3<10);

暂停：

**pause** 10

其他：

写出数据 **write** @varname to outfacename;

读入数据 **read** @varname from infacename;

recve结果 **last\_recve\_ok**

**print**

1. 分支语句

**if** cond\_exp **then**

;

**else**

;

**end**

**switch** cond\_exp

**case** exp

**;;**

**break;**

**default**

**break;**

**end**

**while** cond\_exp

**end;**

1. 运算表达式

加 **+**

减 **-**

乘 **\***

除 **/**

取负数 **-**

取整除法 **//**

取模 **%**

指数（幂） **^**

比较：**< 、>、 <=**、 **>=**、 **==** 、**!=**

与或非： **&&**、**||、！**

1. 函数

type **function** funname(type para1, type para2)

;;;

**end**

1. 事件

**in event** handler(protocol paraname)；

;;;

**end**

1. 接口

**in interface** protocol infacename;

**out interface** protocol outfacename;