模拟编程篇

BY_JC/HC

2022-05-20

VIS：双路电压电流源

V1 版本±50V/±500MA
V2 版本±50V/±1000MA（DEBUG 模式 500MA 以上不可持续 5S 以上）

1、SET-FORCE 指令

并行指令

函数原型 1：void SET_VIS_P(int iPath, unsigned int Mode, double Value, unsigned int Value_Unit, double Clamp_Value, unsigned int Clamp_Unit);

int iPath :第几路多 SITE 施加：1SITE 或 2SITE_A+B 取值范围为 1~4、2SITE 4SITE_A+B 取值范围为 1~2
int Mode : VIS 工作模式 FVMI/FIMV
double Value : 施加值
int Value_Unit : 施加值单位：V: V/MV/UV I:A/MA/UA
double Clamp_Value :钳位值
int Clamp_Unit : 施加钳位值单位：V: V/MV I:A/MA/UA

例:

FVMI

SET_VIS_P(1,FVMI,5,V,20,MA);

释： VIS 通道 1,加 5V 电压,电流钳位 20MA,3V 以上默认为 50V 档位

例:

FIMV

SET_VIS_P(1,FIMV,1,MA,2,V,10);

释： VIS 通道 1,加 1MA 电流,电压钳位 10V

函数原型 2：void SET_VIS_P(int iPath,unsigned int Mode, double Value,unsigned int Value_Unit, double Clamp_Value, unsigned int Clamp_Unit, double Vrange);

int iPath :第几路多 SITE 施加：1SITE 或 2SITE_A+B 取值范围为 1~4、2SITE ，4SITE 取值范围为 1~2
int Mode : VIS 工作模式 FVMI/FIMV
double Value : 施加值
int Value_Unit : 施加值单位：V: V/MV/UV I:A/MA/UA
double Clamp_Value :钳位值
int Clamp_Unit : 施加钳位值单位：V: V/MV I:A/MA/UA
int Vrange ： Vrange 是一个用于选择施加指定量程

Vrange=0 时根据 Value 的值自动选择加压量程。

Vrange>0 时：在 FIMV 方式根据 Vrange 选择加流量程，单位与指令中的 Value_Unit 一致。在 FVMI 方式根据 Vrange 选择加压量程，单位与指令中的 Value_Unit 一致。本指令在 FVMI 方式下量程设置与 SET_VIS_RANGE 指令设置无关

例: **FVMI**

SET_VIS_P(1,FVMI,5,V,20,MA,20);

释： VIS 通道 1,加 5V 电压,电流钳位 20MA,指定 20V 电压档

例: **FIMV**

SET_VIS_P(1,FIMV,15,MA,2,V,20);

释： VIS 通道 1,加 1MA 电流,电压钳位 10V,指定 20MA 电流档

单测指令

函数原型： void SET_VIS 或 VISN(int iMeasNo, unsigned int Mode, double Value, unsigned int Value_Unit, double Clamp_Value, unsigned int Clamp_Unit);

int iMeasNo : 设置第几路施加，范围 1-8
int Mode : VIS 工作模式 FVMI/FIMV
double Value : 施加值
int Value_Unit : 施加值单位：V: V/MV/UV I:A/MA/UA
double Clamp_Value :钳位值
int Clamp_Unit : 施加钳位值单位：V: V/MV I:A/MA/UA

例: **FVMI**

SET_VIS 或 VISN(7,FVMI,5,V,20,MA);

释： VIS 通道 7,加 5V 电压,电流钳位 20MA

例: **FIMV**

SET_VIS 或 VISN(1,FIMV,1,MA,10,V);

释： VIS 通道 1,加 1MA 电流,电压钳位 10V

2、SET-MEASURE

并行指令

函数原型：double VIS_MEASURE_P (int iPath, unsigned int tDelay);

int iPath: 第几路多 SITE 测试, 1SITE 或 2SITE_A+B 取值范围为 1~4、2SITE 或 4SITE 取值范围为 1~2。
int tDelay: 测量延时，单位毫秒

例：VIS_MEASURE_P(1,5);
释：VIS 通道 1 测量,5MS 测量延迟

单测指令

函数原型：VIS 或 VISN_MEASURE (int nVis,unsigned int tDelay)

Int nVis : 指定测量第几路, 范围 1-8。
int tDelay : 测量延时, 单位毫秒
返回测量值单位：电压值为 V,电流值为 A

例：**FVMI**

```
SET_VIS_P(1,FVMI,5,V,20,MA);
VIS_MEASURE_P(1,5);
    for(j=0;j<iNum;j++)
MeasureI[j]=GET_DATA(j);
```

FIMV

```
SET_VIS_P(1,FIMV,100,MA,2,V);
VIS_MEASURE_P(1,5);
    for(j=0;j<iNum;j++)
MeasureV[j]=GET_DATA(j);
```

3、CONDITIONS 指令

并行指令

函数原型：void CONDITIONS_VIS_P (int iPath, unsigned int Mode, double Value, unsigned int Value_Unit, double Clamp_Value, unsigned int Clamp_Unit);

int iPath: 第几路多 SITE 测试：1SITE 或 2SITE_A+B 取值范围为 1~4、2SITE 或 4SITE 取值范围为 1~2
int Mode : VIS 工作模式 FVMI/FIMV
double Value : 施加值
int Value_Unit : 施加值单位：V: V/MV/UV I:A/MA/UA
double Clamp_Value : 钳位值

int Clamp_Unit : 施加箝位值单位：V: V/MV　I:A/MA/UA

例：CONDITIONS_VIS_P(1,FVMI,1,V,2,MA);
释：VIS 通道 1,加压 1V,钳位 2MA,只设条件不输出

单测指令

函数原型：void CONDITIONS _VISN (int iMeasNo, unsigned int Mode, double Value, unsigned int Value_Unit, double Clamp_Value, unsigned int Clamp_Unit);
int iMeasNo : 设置第几路施加，范围 1-8
int Mode : VIS 工作模式 FVMI/FIMV
double Value : 施加值
int Value_Unit : 施加值单位：V: V/MV/UV　I:A/MA/UA
double Clamp_Value : 钳位值
int Clamp_Unit : 施加钳位值单位：V: V/MV I:A/MA/UA

例：CONDITIONS_VISN(8,FVMI,1,V,2,MA);
释：VIS 通道 8,加压 1V,钳位 2MA,只设条件不输出

4、CONDITIONS-MEASURE

并行指令

函数原型：double MEASURE _VIS_P (int iPath, unsigned int tDelay);

int iPath: 第几路多 SITE 测试：1SITE 或 2SITE_A+B 取值范围为 1~4、2SITE 或 4SITE_A+B 取值范围为 1~2
int tDelay : 测量延时, 单位毫秒

例：MEASURE_VIS_P(1,5);
释：输出 CONDITIONS 指令的设置条件,VIS 通道 1 测试,5MS 测量延迟,之后自动断开输出

单测指令

函数原型：MEASURE _VISN (int nVis, unsigned int tDelay) ;

int nVis : 指定测量第几路, 范围 1-8
int tDelay : 测量延时, 单位毫秒

例：MEASURE_VISN(1,5);

释：输出 CONDITIONS 指令的设置条件,VIS 通道 1 测试,5MS 测量延迟,之后自动断开输出

解释：VIS 通道 1,输出上边的 CONDITIONS-FVMI 或 FIMV 测量后断开输出

例 ：**FIMV**

```
CONDITIONS_VIS_P(1,FIMV,10,MA,2,V);
MEASURE_VIS_P(1,5);
for(j=0;j<iNum;j++)
MeasureV[j] = GET_DATA(j);
```

FVMI

```
CONDITIONS_VIS_P(1,FVMI,2,V,10,MA);
MEASURE_VIS_P(1,5);
for(j=0;j<iNum;j++)
MeasureI[j] = GET_DATA(j);
```

LVIS：四路电压电流源

V1 版本±24V/±250MA

V2 版本±24V/±1000MA（DEBUG 模式 500MA 以上不可持续 5S 以上）

1、SET-FORCE 指令

并行指令

函数原型 1：void SET_LVIS_P(int iPath, unsigned int Mode, double Value, unsigned int Value_Unit, double Clamp_Value, unsigned int Clamp_Unit);

int iPath :第几路多 SITE 施加：1SITE 或 2SITE_A+B 取值范围为 1~4、2SITE 4SITE_A+B 取值范围为 1~2

int Mode : LVIS 工作模式 FVMI/FIMV

double Value : 施加值

int Value_Unit : 施加值单位：V: V/MV/UV I:A/MA/UA

double Clamp_Value :钳位值

int Clamp_Unit : 施加钳位值单位：V: V/MV I:A/MA/UA

例: **FVMI**

```
SET_LVIS_P(1,FVMI,5,V,20,MA);
```

释：LVIS 通道 1,加 5V 电压,电流钳位 20MA,3V 以上默认为 24V 档位

例: **FIMV**

SET_LVIS_P(1,FIMV,1,MA,2,V,10);

释： LVIS 通道 1,加 1mA 电流,电压钳位 10V

函数原型 2：void SET_LVIS_P(int iPath,unsigned int Mode, double Value,unsigned int Value_Unit, double Clamp_Value, unsigned int Clamp_Unit, double Vrange);

int iPath :第几路多 SITE 施加：1SITE 或 2SITE_A+B 取值范围为 1~4、2SITE ，4SITE 取值范围为 1~2

int Mode : LVIS 工作模式 FVMI/FIMV

double Value : 施加值

int Value_Unit : 施加值单位：V: V/MV/UV I:A/MA/UA

double Clamp_Value :钳位值

int Clamp_Unit : 施加钳位值单位：V: V/MV I:A/MA/UA

int Vrange ： Vrange 是一个用于选择施加指定量程

Vrange=0 时根据 Value 的值自动选择加压量程。

Vrange>0 时：在 FIMV 方式根据 Vrange 选择加流量程，单位与指令中的 Value_Unit 一致。在 FVMI 方式根据 Vrange 选择加压量程，单位与指令中的 Value_Unit 一致。本指令在 FVMI 方式下量程设置与 SET_LVIS_RANGE 指令设置无关

例: **FVMI**

SET_LVIS_P(1,FVMI,5,V,20,MA,20);

释： LVIS 通道 1,加 5V 电压,电流钳位 20MA,指定 20V 电压档

例: **FIMV**

SET_LVIS_P(1,FIMV,15,MA,2,V,20);

释： LVIS 通道 1,加 1mA 电流,电压钳位 10V,指定 20MA 电流档

单测指令

函数原型：void SET_LVIS(int iMeasNo, unsigned int Mode, double Value, unsigned int Value_Unit, double Clamp_Value, unsigned int Clamp_Unit);

int iMeasNo : 设置第几路施加, 范围 1-8

int Mode : LVIS 工作模式 FVMI/FIMV

double Value : 施加值

int Value_Unit : 施加值单位：V: V/MV/UV I:A/MA/UA

double Clamp_Value :钳位值

int Clamp_Unit : 施加钳位值单位：V: V/MV I:A/MA/UA

例: **FVMI**

SET_LVIS(7,FVMI,5,V,20,MA);

释： LVIS 通道 7,加 5V 电压,电流钳位 20MA

例: **FIMV**

SET_LVIS(1,FIMV,1,MA,10,V);

释： LVIS 通道 1,加 1MA 电流,电压钳位 10V

2、SET-MEASURE

并行指令

函数原型：double LVIS_MEASURE_P (int iPath, unsigned int tDelay);

int iPath: 第几路多 SITE 测试, 1SITE 或 2SITE_A+B 取值范围为 1~4、2SITE 或 4SITE 取值范围为 1~2。

int tDelay: 测量延时，单位毫秒

例：LVIS_MEASURE_P(1,5);

释：LVIS 通道 1 测量,5MS 测量延迟

单测指令

函数原型：LVIS_MEASURE (int nLVis,unsigned int tDelay)

Int nLVis : 指定测量第几路, 范围 1-8

int tDelay : 测量延时, 单位毫秒

返回测量值单位：电压值为 V,电流值为 A

例：**FVMI**

```
SET_LVIS_P(1,FVMI,5,V,20,MA);
LVIS_MEASURE_P(1,5);
    for(j=0;j<iNum;j++)
MeasureI[j]=GET_DATA(j);
```

FIMV

```
SET_LVIS_P(1,FIMV,100,MA,2,V);
LVIS_MEASURE_P(1,5);
    for(j=0;j<iNum;j++)
MeasureV[j]=GET_DATA(j);
```

3、CONDITIONS 指令

并行指令

函数原型：void CONDITIONS_LVIS_P (int iPath, unsigned int Mode, double Value, unsigned int Value_Unit, double Clamp_Value, unsigned int Clamp_Unit);

int iPath: 第几路多 SITE 测试：1SITE 或 2SITE_A+B 取值范围为 1~4、2SITE 或 4SITE 取值范围为 1~2
int Mode : LVIS 工作模式 FVMI/FIMV
double Value : 施加值
int Value_Unit : 施加值单位：V: V/MV/UV I:A/MA/UA
double Clamp_Value : 钳位值
int Clamp_Unit : 施加箝位值单位：V: V/MV I:A/MA/UA

例：CONDITIONS_LVIS_P(1,FVMI,1,V,2,MA);
释：LVIS 通道 1,加压 1V,钳位 2MA,只设条件不输出

单测指令

函数原型：void CONDITIONS _LVIS (int iMeasNo, unsigned int Mode, double Value, unsigned int Value_Unit, double Clamp_Value, unsigned int Clamp_Unit);
int iMeasNo : 设置第几路施加，范围 1-8
int Mode : LVIS 工作模式 FVMI/FIMV
double Value : 施加值
int Value_Unit : 施加值单位：V: V/MV/UV I:A/MA/UA
double Clamp_Value : 钳位值
int Clamp_Unit : 施加钳位值单位：V: V/MV I:A/MA/UA

例：CONDITIONS_LVIS(8,FVMI,1,V,2,MA);
释：LVIS 通道 8,加压 1V,钳位 2MA,只设条件不输出

4、CONDITIONS-MEASURE

并行指令

函数原型：double MEASURE _LVIS_P (int iPath, unsigned int tDelay);

int iPath: 第几路多 SITE 测试：1SITE 或 2SITE_A+B 取值范围为 1~4、2SITE 或 4SITE_A+B 取值范围为 1~2

int tDelay : 测量延时, 单位毫秒

例：MEASURE_LVIS_P(1,5);
释：输出 CONDITIONS 指令设置条件,LVIS 通道 1 测试,5MS 测量延迟,之后自动断开输出

单测指令

函数原型：MEASURE _LVIS (int nLVis, unsigned int tDelay) ;

int nLVis : 指定测量第几路, 范围 1-8
int tDelay : 测量延时, 单位毫秒

例：MEASURE_LVIS(1,5);
释：输出 CONDITIONS 指令设置条件,LVIS 通道 1 测试,5MS 测量延迟,之后自动断开输出

例 ：**FIMV**

```
CONDITIONS_LVIS_P(1,FIMV,10,MA,2,V);
MEASURE_LVIS_P(1,5);
 for(j=0;j<iNum;j++)
MeasureV[j] = GET_DATA(j);
```

FVMI

```
CONDITIONS_LVIS_P(1,FVMI,2,V,10,MA);
MEASURE_LVIS_P(1,5);
 for(j=0;j<iNum;j++)
MeasureI[j] = GET_DATA(j);
```

PVIS(VI5A)：单路功率电压电流源

±50V/±10A（500MA 以上持续 120MS 左右自动断开输出）

1、SET_FORCE 指令

并行指令

函数原型： void SET_VI5A_P(int iPath, unsigned int Mode, double Value, unsigned int Value_Unit, double Clamp_Value, unsigned int Clamp_Unit, int iKac, int iKzj,int Time_Value);

int iPath: 第几路功率源,多 SITE 施加：1SITE 或 2SITE_A+B 取 值范围为 1~2、2SITE 或 4SITE 取值范围为 1。
int Mode : PVIS 工作模式 FVMI/FIMV
double Value : 施加值(常数或函数)
int Value_Unit : 施加值单位： V: V/MV/UV I:A/MA/UA
double Clamp_Value : 钳位值
int Clamp_Unit : 施加钳位值单位： V: V/MV I:A/MA/U
int iKac： 保留, 设为 0
int iKz ： iKz >3 时：为量程选择 在加流方式根据 iKz 选择加流量程, 单位与指令中的 Value_Unit 一致。 在加压方式根据 iKz 选择加压量程，单位与指令中的 Value_Unit 一致。本指令在加压方式下量程设置与 SET_VI5A_RANGE 指令设置无关
int Time_Value：施加的持续时间, 除 0 表示直流外, 其它毫秒为单位的时间值

FVMI

例 1： SET_VI5A_P(1,FVMI,5,V,20,MA,0,0,0);
释：PVIS 通道 1,施加 5V 电压,电流钳位 20MA

例 2：SET_VI5A_P(1,FVMI,5,V,20,MA,0,10,500);
释：PVIS 通道 1,施加 5V,钳位 20MA,指定 10V 量程,持续 500MS

FIMV

例 3：SET_VI5A_P(1,FIMV,5,MA,2.0,V,0,0,0);
释：PVIS 通道 1,施加 5MA 电流,电流钳位 2V

例 4：SET_VI5A_P(1,FIMV,5,MA,2.0,V,0,10,500);
释：PVIS 通道 1,施加 5MA 电流,电流钳位 2V,指定 10MA 电流量程,持续 500MS

单测指令

函数原型:SET_VI5A (int iPath, unsigned int Mode, double Value, unsigned int Value_Unit, double Clamp_Value, unsigned int Clamp_Unit, int iKac, int iKzj,int Time_Value);

int iPath: 指定第几路功率源,取值范围为 1~4。
int Mode : PVIS 工作模式 FVMI/FIMV
double Value : 施加值
int Value_Unit : 施加值单位： V: V/MV/UV I:A/MA/UA
double Clamp_Value : 钳位值
int Clamp_Unit : 施加钳位值单位： V: V/MV I:A/MA/U

int iKac： 保留, 设为 0
int iKz： 保留, 设为 0
int Time_Value：施加的持续时间, 除 0 表示直流外, 其它毫秒为单位的时间值。

FVMI

例 1： SET_VI5A(1,FVMI,5,V,20,MA,0,0,0);
释：PVIS 通道 1,施加 5V 电压,电流钳位 20MA

例 2：SET_VI5A(1,FVMI,5,V,20,MA,0,0,500);
释：PVIS 通道 1,施加 5V,钳位 20MA,持续 500MS

FIMV

例 3：SET_VI5A(1,FIMV,5,MA,2.0,V,0,0,0);
释：PVIS 通道 1,施加 5MA 电流,电流钳位 2V

例 4：SET_VI5A(1,FIMV,5,MA,2.0,V,0,0,500);
释：PVIS 通道 1,施加 5MA 电流,电流钳位 2V,持续 500MS

2、SET-MEASURE

并行指令

函数原型：double VI5A_MEASURE_P (int iPath, unsigned int tDelay);

int iPath: 第几路多 SITE 测试：1SITE 或 2SITE_A+B 取值范围为 1~2、2SITE 或 4SITE 取值范围为 1
int tDelay : 测量延时, 单位毫秒

例：VIN5A_MEASURE_P(1,5);
释：PVIS 通道 1 测量,5MS 测量延迟

FVMI

```
SET_VI5A_P(1,FVMI,5,V,2.0,MA,0,0,0);
VIN5A_MEASURE_P(1,5);
   for(j=0;j<iNum;j++)
MeasureI[j]=GET_DATA(j);
```

FIMV

```
SET_VI5A_P(1,FIMV,5,MA,2.0,V,0,0,0);
VIN5A_MEASURE_P(1,5);
   for(j=0;j<iNum;j++)
MeasureV[j]=GET_DATA(j);
```

单测指令

函数原型：double VIS5A_MEASURE_P (int iPath, unsigned int tDelay);

int iPath:指定第几路功率源,取值范围为 1~4
int tDelay:测量延时, 单位毫秒

例：VIN5A_MEASURE(1,5);
释：PVIS 通道 1 测量,5MS 测量延迟

3、CONDITIONS

并行指令

函数原型：CONDITIONS_VI5A_P(int iPath, unsigned int Mode, double Value,
unsigned int Value_Unit, double Clamp_Value, unsigned int Clamp_Unit， int iKzj);

int iPath: 第几路功率源,多 SITE 施加：1SITE 或 2SITE_A+B 取 值范围为 1~2、2SITE 或 4SITE 取值范围为 1。
int Mode : PVIS 工作模式 FVMI/FIMV
double Value : 施加值
int Value_Unit : 施加值单位：V: V/MV/UV I:A/MA/UA
double Clamp_Value : 钳位值
int Clamp_Unit : 施加钳位值单位：V: V/MV I:A/MA/U
int iKz: 保留, 设为 0

例：

FVMI

CONDITIONS_VI5A_P(1,FVMI,1,V,100,MA,0);

FIMV

CONDICTIONS_VI5A_P(1,FIMV,100,MA,1,V,0);

单测指令

函数原型：CONDITIONS_VIN5A(int iPath, unsigned int Mode, double Value,

unsigned int Value_Unit, double Clamp_Value, unsigned int Clamp_Unit， int iKzj);

int iPath: 第几路功率源,取 值范围为 1~4
int Mode : PVIS 工作模式 FVMI/FIMV
double Value : 施加值
int Value_Unit : 施加值单位：V: V/MV/UV I:A/MA/UA
double Clamp_Value : 钳位值
int Clamp_Unit : 施加钳位值单位：V: V/MV I:A/MA/U
int iKz: 保留, 设为 0

例：

FVMI

```
CONDITIONS_VIN5A(1,FVMI,1,V,100,MA,0);
```

FIMV

```
CONDITIONS_VIN5A(1,FIMV,100,MA,1,V,0);
```

4、CONDITIONS-MEASURE

并行指令

函数原型：double MEASURE _VIN5A_P (int iPath, unsigned int tDelay);

int iPath: 第几路多 SITE 测试：1SITE 或 2SITE_A+B 取值范围为 1~4、2SITE 或 4SITE 取值范围为 1~2
int tDelay : 测量延时, 单位毫秒

例：**FVMI**

```
CONDITIONS_VI5A_P(1,FVMI,1,V,100,MA,0);
MEASURE_VIN5A_P(1,10);
  for(j=0;j<iNum;j++)
MeasureI[j] = GET_DATA(j);
```

FIMV

```
CONDITIONS_VI5A_P(1,FIMV,100,MA,1,V,0);
MEASURE_VIN5A_P(1,10);
  for(j=0;j<iNum;j++)
MeasureV[j] = GET_DATA(j);
```

单行指令

函数原型：double MEASURE _VIN5A (int iPath, unsigned int tDelay);

int iPath: 范围为 1~4
int tDelay : 测量延时, 单位毫秒

例：**FVMI**

```
CONDITIONS_VIN5A(1,FVMI,1,V,100,MA,0);
MEASURE_VIN5A(1,10);
   for(j=0;j<iNum;j++)
MeasureI[j] = GET_DATA(j);
```

FIMV

```
CONDITIONS_VIN5A(1,FIMV,100,MA,1,V,0);
MEASURE_VIN5A(1,10);
   for(j=0;j<iNum;j++)
MeasureV[j] = GET_DATA(j);
```

以下篇幅指令仅以例程形式说明，不展示函数原型

DVM:四路直流数字电压表

±50V 电压测量

1、常规测量

并行指令：DVM_MEASURE_P(1,5,V,10);
单测指令：DVM_MEASURE(1,5,V,10);
解释：DVM 通道 1 测量,5V 电压档位,10mS 测量延迟

例：

```
DVM_MEASURE_P(1,5,V,10);
for(j=0;j<iNum;j++)
MeasureV[j] = GET_DATA(j);//单位 V
```

2、DIFF 测量

并行指令：DVM_DIFF_MEASURE_P(2,5,V,10);
单测指令：DVM_DIFF_MEASURE(2,5,V,10);
解释：DVM 通道 2,5V 档位测量,10mS 前后的电压差

TMU：四路时间测量单元

10M 内，20nS~40S，TG±10V

并行指令_P
单测指令无_P

1、频率测量

例：

MEASURE_PERIOD_P(T1,2,V,5,10,0);

for(j=0;j<iNum;j++)

MEASURE_F[j]=GET_DATA(j);//单位 HZ

解释：TMU 通道 1,TG2V,测 5 个周期,10mS 测量延迟,0=慢速测量,1=快速测量

2、周期测量

例：

MEASURE_TIME_P(T1,POS,2,V,T1,POS,2,V,10,1);

或 MEASURE_TIME_P(T1,NEG,2,V,T1,NEG,2,V,10,1);

for(j=0;j<iNum;j++)

MEASURE_T[j]=GET_DATA(j);//单位 S

解释：TMU 通道 1,上升沿或下降沿 TG2V,10mS 测量延迟,0=慢速测量,1=快速测量

3、高电平脉宽测量

例：

MEASURE_TIME_P(T1,POS,2,V,T1,NEG,2,V,10,1);

for(j=0;j<iNum;j++)

MEASURE_H[j]=GET_DATA(j);//单位 S

解释：TMU 通道 1,上升沿 TG2V,下降沿 TG2V,10mS 测量延迟,0=慢速测量,1=快速测量

4、低电平脉宽测量

例：

MEASURE_TIME_P(T1,NEG,2,V,T1,POS,2,V,10,1);

for(j=0;j<iNum;j++)

MEASURE_L[j]=GET_DATA(j);//单位 S

解释：TMU 通道 1,下降沿 TG2V,上升沿 TG2V,10mS 测量延迟,0=慢速测量,1=快速测量

5、单次触发时间测量,该指令适用并行测试,无单测指令

例：①测一个电源的上升沿

TMU_TIME_SET(T1,POS,0.5,V,T1,POS,8.5,V,1);

TMU_CONNECT (20); //Delay20mS,

SET_VIS_P(1,FVMI,9,V,20,MA); //CONNECT TO T1 触发点

TMU_READ_TIME (10);//数据读取前的等待时间 10mS

for(j=0;j<iNum;j++)

MeasureU[j] =GET_DATA(j);

②测两个通道的传输延迟

TMU_TIME_SET(T1,POS,4,V,T2,NEG,10,V,0);//T1 通道上升，T2 通道下降
或 TMU_TIME_SET(T1,POS,4,V,T2,POS,10,V,0);//T1 通道上升，T2 通道上升
或 TMU_TIME_SET(T1,NEG,4,V,T2,POS,10,V,0);//T1 通道下降，T2 通道上升
或 TMU_TIME_SET(T1,NEG,4,V,T2,NEG,10,V,0);//T1 通道下降，T2 通道下降
TMU_CONNECT(20);//Delay20mS,
SET_VIS_P(2,FVMI,5,V,20,MA);//CONNECT TO T1 触发点
Delay(20);
TMU_READ_TIME(10);//数据读取前的等待时间 10mS
for(j=0;j<iNum;j++)
MeasureD[j] =GET_DATA(j);

HVIS:单路程控高压源

1000V/10MA

SET_HVIS (FVMI, 500,V,20,UA);
解释：设置高压 HVIS_FVMI 模式 500V,20UA 电流档

SET_HVIS (FVMI, 500,V,0,UA,800,V);
解释：设置高压 HVIS_FVMI 模式 500V,0UA 电流档,800V 档位

SET_HVIS (FIMV, 250,UA,700,V);
解释：设置高压 HVIS_FIMV 模式 250uA,700V 钳位电压

SET_HVIS_ZERO ();
解释：保留 HVIS 输出条件设置,仅仅对外输出 0

例：测量 MOS 耐压 BVDSS,IDSS

SET_HVIS(FIMV,250,UA,900,V);
Delay(50);
Vdss[0] =MEASURE_HVIS();
SET_HVIS_ZERO ();

SET_HVIS(FVMI,600,V,50,UA);
Delay(50);
Idss[0] =MEASURE_HVIS()*1e6;
CLEAR_HVIS();

AS：双路交流电压源

1HZ~200KHZ

正弦波

SET_AS_P(1, 5, V, 1000);

解释：AS 通道 1,输出 5V 均方根值,1000HZ 的正弦波

方波

SET_AS_SQUARE_P(1,5,V,1000);

解释：AS 通道 1,低 0V,高 5V,频率 1KHZ 的方波,AS 方波低不过 0 点

例：测试音频功放 IC 的波形输出,失真度 THD,平均值(均方根值 RMS),功率 PO

主函数上边需增加如下额外函数:

```
// Nunmax
double cos_part[512][512]={0};
double sin_part[512][512]={0};
double fValue[600][4],Vrms[4];
bool    flag=true;

void CLOSE_DIFF_CHANNEL()
{
  CLOSE_RELAY("97,98");//97 98 (AVM1,3 通道) ,102 103(AVM2,4 通道)
}
void SIN_COS()   //初始化 DFT 计算时 SIN 和 COS 函数值
{
   int   i,n,N=512;
      for(i=0; i<N; i++)
      {
         for(n=0; n<N; n++)
         {
               cos_part[i][n] = cos(2*3.1415926535/N*i*n);
               sin_part[i][n] = sin(2*3.1415926535/N*i*n);
         }
      }
}

void thd_start_adrr(double (*fValue)[4],int *adrr_start,int math_long)
```

```
{
   double V1[200];
   int    iNum= WORD(GET_DATA(5));
      for(int  j=0;j<iNum;j++)
   {
        for(int i=0;i<math_long;i++)
              V1[i]=fValue[i][j];
        if((V1[0]>0)&&(V1[int(20)]>0))
        {
              for(i=1;i<math_long;i++)
              {
                    if((V1[i+1]<V1[i])&&(V1[i+1]>0))
                          break;
              }
        }
        if((V1[0]>0)&&(V1[10]<0))
        {
              for(i=1;i<math_long;i++)
              {
                    if((V1[i+1]>V1[i])&&(V1[i+1]<0))
                          break;
              }
        }
        if(V1[0]<0)
        {
              for(i=1;i<math_long;i++)
              {
                    if((V1[i+1]<V1[i])&&(V1[i+1]>0))
                          break;
              }
        }
        adrr_start[j]=i;
   }
}
```

主函数体内 THD /VRMS /PO 测量指令：

```
   SET_AS_P(1,200,MV,1000);//AS 产生输入幅度 200mV RMS、1KHZ 正弦波
   Delay(60);
//RMS
   MAT_DVM_MEASURE_P(1,5,V,5,(Nunmax+start_long),100000,fValue);
   for(j=0;j<iNum;j++)
```

```
    Vrms[j]=VRMS(fValue,Nunmax,j);
    if(!SHOW_RESULT("VRMS_AB",Vrms,"V",3.2,2.5,6))
        return;
  //THD%
    thd_start_adrr(fValue,adrr_start,start_long);
    DFT(fValue, Nunmax, cos_part,sin_part,&thd[0],adrr_start);
    if(!SHOW_RESULT("THD_%",thd,"%",0.5,0,5))
     return;
  //THD_dB
    for(j=0;j<iNum;j++)
    thd[j]=20*log10(thd[j]/100);

    if(!SHOW_RESULT("THD_dB_AB",thd,"dB",-28,No_LoLimit,6))
      return;
  //PO
      for(j=0;j<iNum;j++)
      PO[j]=Vrms[j]*Vrms[j]/8;//8 欧姆负载
        if(!SHOW_RESULT("Po_AB",PO,"W",2,0,6))
        return;

    SET_AS_P(1,0,V,1000);
```

其他指令以及应用场合

----------继电器指令----------

特殊说明： **①继电器_P 指令自动映射关系为 1-49,2-50,3-51 直到 48-96**
②单测试头模式下继电器操作无自动映射,一般不写_P 指令 SET_RELAY_P("1"),而是分开写实际动作的继电器,SET_RELAY("1")
③继电器指令内的参数为字符串,如需继电器控制位为变量必须进行格式转换

例： 变量操作继电器

```
CString str1;//定义字符串
str1.Format("%d,%d,%d,%d",A,B,C,D); //变量转换成字符串格式
Delay(1);
CLOSE_RELAY(str1);        //操作字符串
Delay(5);
CLEAR_RELAY(str1);        //操作字符串
Delay(5);
```

1、继电器 SET_RELAY 指令

例：闭合 RELAY 1-5

```
SET_RELAY("1,2,3,4,5");
SET_RELAY("1-5");
SET_RELAY("1,2,3-5");
```

解释：SET_RELAY 指令会清掉之前的继电器状态

2、继电器 CLOSE_RELAY 指令

例：闭合 RELAY 1-5

```
CLOSE_RELAY("1,2,3,4,5");
CLOSE_RELAY("1-5");
CLOSE_RELAY("1,2,3-5");
```

解释：CLOSE_RELAY 指令不会清掉之前的继电器状态

3、继电器清除指令

```
SET_RELAY("0");
CLEAR_RELAY("1,2,3,4,5");
```

4、继电器分组指令（2022 年 5 月前出厂的设备需要软件升级方可支持此指令）

①分 2 组，双工位并行操作指令

SET_P2S_RELAY(int **X**,**SPACE**,int **a**,int **b**,int **c**,int **d**,int **e**....);//最多容纳 24 个控制位号变量

解释：变量 X=继电器平均分组的组与组之间的 CBIT 位号间隔数，a,b,c,d,e....=CBIT 位号变量，CBIT 变量为大于 0 的整数,配套指令 CLOSE_P2S_RELAY(),CLEAR_P2S_RELAY()

例：SET_P2S_RELAY(10,SPACE,1,2,3,4,5);

分两组继电器,组间距为 10，执行此指令实际闭合的继电器为 K1-K5,K11-K15,

若 X=0，则实际只闭合 K1-K5

②分 4 组，四工位并行操作指令

SET_P4S_RELAY(int **X**,**SPACE**,int **a**,int **b**,int **c**,int **d**,int **e**....);//最多容纳 24 个控制位号变量

解释：变量 X=继电器平均分组的组与组之间的 CBIT 位号间隔数，a,b,c,d,e....=CBIT 位号变量,为大于 0 的整数,配套指令 CLOSE_P4S_RELAY(),CLEAR_P4S_RELAY()

例：SET_P4S_RELAY(10,SPACE,1,2,3,4,5);

分四组继电器,组间距为 10,执行此指令实际闭合的继电 K1-K5,K11-K15,K21-K25,K31-K35 若 X=0，则实际只闭合 K1-K5

注: 上述①、②指令在并行测试中此指令不受 FAIL_STOP 控制，如特殊情况下 SITE 之间有影响则需单独操作,括号内 CBIT 位号为变量,不支持字符串型指令格式,需要一一列出所需的位号

③分 1~4 组，最大 4 组，单独组操作指令(多用于 CP 中多 SITE 测试 TRIM FUSE)

SET_GROUP_RELAY(int Y,GP,int X,SPACE,int a,int b,int c,int d,int e....);

解释：Y=继电器分组号,X=继电器平均分组的组与组之间的间隔数，a,b,c,d,e.....=CBIT 位号
Y 取值范围为 0-4，Y=1,为第一组,Y=2 为第二组 ,Y=3 为第三组,Y=4 为第四组,
Y=0 为复位系统所有用户电器
a,b,c,d,e....=CBIT 位号变量，为大于 0 的整数
若 X=0,则只执行第一组继电器动作
相关配套指令有 CLOSE_GROUP_RELAY(),CLEAR_GROUP_RELAY()

例: SET_GROUP_RELAY(1,GP,20,SPACE,1,2,3,4);
闭合第 1 组的继电器 K1,K2,K3,K4

SET_GROUP_RELAY(2,GP,20,SPACE,1,2,3,4);
闭合第 2 组的继电器 K21,K22,K23,K24

SET_GROUP_RELAY(3,GP,20,SPACE,1,2,3,4);
闭合第 3 组的继电器 K41,K42,K43,K44

SET_GROUP_RELAY(4,GP,20,SPACE,1,2,3,4);
闭合第 4 组的继电器 K61,K62,K63,K64

SET_GROUP_RELAY(0,GP,20,SPACE,1,2,3,4);
继电器状态总清零

SET_GROUP_RELAY(1,GP,0,SPACE,1,2,3,4);
SET_GROUP_RELAY(2,GP,0,SPACE,1,2,3,4);
SET_GROUP_RELAY(3,GP,0,SPACE,1,2,3,4);
SET_GROUP_RELAY(4,GP,0,SPACE,1,2,3,4);
上面四条指令一样，只闭合第 1 组继电器 K1,K2,K3,K4

例：4SITE 测试中烧熔丝 TRIM FUSE 例程

```
            SET_VIS_P(1,FVMI,5,V,20,MA);
            DVM_MEASURE_P(1,5,V,10);
              for(j=0;j<iNum;j++)
             Vref[j]=GET_DATA(j);
//trimming
          for(j=0;j<iNum;j++)
  {
          if(2.45<Vref[j]<=2.48)
        {
         CLOSE_GROUP_RELAY(j+1,GP,8,SPACE,1,2,3);//GP 号与 SITE 一一对应
         Delay(5);
         CLEAR_GROUP_RELAY(j+1,GP,8,SPACE,1,2,3);
        }

         if(2.48<Vref[j]<=2.50)
        {
         CLOSE_GROUP_RELAY(j+1,GP,8,SPACE,1,2,4);
         Delay(5);
         CLEAR_GROUP_RELAY(j+1,GP,8,SPACE,1,2,4);
        }

        if(2.50<Vref[j]<=2.53)
        {
         CLOSE_GROUP_RELAY(j+1,GP,8,SPACE,2,3,5,6);
         Delay(5);
         CLEAR_GROUP_RELAY(j+1,GP,8,SPACE,2,3,5,6);
        }

 }
```

----------采样读取指令----------

适用场合：根据测试具体要求，合理优化测试时间

1、采样

SET_SAMPLING_NUM(0);//全局采样数设置 30
SET_SAMPLING_NUM(1);//全局采样数设置 60
SET_SAMPLING_NUM(2);//全局采样数设置 120,与无该指令相同

说明：括号内采样最小数为 10,最大 120,全局有效直到下一次设置为止

2、读取

READ_AD_MODE(0);
说明:软件读取,用于细测,系统默认为 0,全局有效直到下一次设置为止
READ_AD_MODE(1);
说明:硬件读取,用于粗测,一般用于测量 500mV 以上较高电平,uA 以上级电流,全局有效直到下一次设置为止

----------清零指令----------

1、并行-指定通道清零,先置 0 后断开

```
CLEAR_VIS_P(1);
CLEAR_LVIS_P(1);
CLEAR_VIN5A_P(1);
```

2、单测-指定通道清零,先置 0 后断开

```
CLEAR_VISN(1);
CLEAR_LVIS(1);
CLEAR_VIN5A(1);
CLEAR_HVIS();
```

3、系统总清

```
CLEAR_ALL();
```

----------设限指令----------

例 1：

```
if(!SHOW_RESULT("ICC1",ICC_AB,"mA",25,10,5))
return;
```

说明："ICC1"=标识符,传递的实体变量为 ICC_AB,显示在 LOGO 中的名字
ICC_AB=实际测试结果
"mA"=经过外部换算后的单位标识
25=SPEC 区间上限
10=SPEC 区间下限
5=软 BIN 号

例 2：

```
if(!SHOW_RESULT("ICC1",ICC_AB,MA,25,10,5))
return;
```

说明："ICC1"=标识符,传递的实体变量为 ICC_AB,显示在 LOGO 中的名字
ICC_AB=实际测试结果
MA=系统内部运算输出的实际单位，不需要外部换算
25=SPEC 区间上限
10=SPEC 区间下限
5=软 BIN 号

例 3：

```
if(!SHOW_RESULT("ICC1",ICC_AB,MA,No_UpLimit,No_LoLimit,5))
return;
```

说明："ICC1"=标识符,传递的实体变量为 ICC_AB,显示在 LOGO 中的名字
ICC_AB=实际测试结果
MA=系统内部运算输出的实际单位，不需要外部换算
No_UpLimit=SPEC 区间不设上限
No_LoLimit=SPEC 区间不设下限
5=软 BIN 号

例 4：

```
if(!SHOW_RESULT("ICC1",ICC_AB,MA,25,-20,5))
 {
  POWER_OFF();//自定义失效返回前的下电函数,一般用于带大电容的放电操作
  return;
```

}

----------程序结尾 PASS_BIN 指令----------

SET_BIN(0,1);

----------测试头数量声明指令----------

SET_COMMAND(1,1);//单测试头模式

SET_COMMAND(1,2);//双测试头模式,系统默认双测试头模式

----------换切电流档不掉电指令----------

FVMI_RANGE_NOBREAK();//用在切换之前

----------SET_RANGE 指令----------

SET_VIS_RANGE(1,FVMI,20);

SET_LVIS_RANGE(1,FVMI,20);

SET_VI5A_RANGE(1,FVMI,20);

说明：仅用在 FVMI 模式下,全程设置电压量程 1=通道,20=电压值

----------扫描监控----------

适用场合：根据输入端电压电流的渐进增加或减小,监控输入或输出端电平或电流的突变点

例程：输入电压初始 2V,然后以 50mV 步进缓慢增加,检测输入端电流大于 5mA 小于 100mA 的时刻输入电压的施加值

```
    INIT_SEARCH_D();//扫描测试的初始指令
  for(double V_temp=2;V_temp<=8;V_temp+=0.05)
{
  SET_VIS_P(1,FVMI,V_temp,V,20,MA);
  VIS_MEASURE_P(1,5);
  for(j=0;j<iNum;j++)
  MeasureI[j]=GET_DATA(j)*1e3;

  if(COMP_SEARCH_D(V_temp,0,MeasureI,100,5))//MeasureI 大于 5MA 小于 100MA 时跳出
    break;
}
  READ_SEARCH_R(1);//回读跳出时"测量"的电流值
  for(j=0;j<iNum;j++)
  MeasureI[j]=GET_DATA(j);
  if(!SHOW_RESULT("MeasureI",MeasureI,"V",100,5,7))
```

```
return;
READ_SEARCH_F(1);//回读跳出时"施加"的电压值
for(j=0;j<iNum;j++)
VST[j]=GET_DATA(j);
if(!SHOW_RESULT("VST",VST,"V",6.5,4.9,7))
return;
```

----------PASS_BIN 细分多档----------

适用场合：需要针对某个或多个参数,进行细分档位,良品复合分档

例程：根据 VCS 的测试结果,分两个良品档

```
int n,test_sign;
int    fbin[4];

for(j=0;j<iNum;j++)
  {
        test_sign=1;
    if((vcs[j]<=252.5)&&(vcs[j]>237.5))//A 档
        test_sign=1;

    if((vcs[j]<=268.3)&&(vcs[j]>252.5))//B 档
        test_sign=2;

     switch (test_sign)
     {
          case 1:
               fbin[j]=1;       break;//BIN1
          case 2:
               fbin[j]=2;       break;//BIN2
     }
  }
  for(n=1;n<=iNum;n++)
  SET_BIN(n,fbin[n-1]);
```

----------获取当前有效 SITE----------

适用场合：用于单资源并行测试中的串行处理,或 CP 测试中因多 SITE 共电源等情况相互影响进行的串行处理

例：以单路高压源 HVIS 为例,4SITE 测试中串行处理,测试 VDSS,IDSS

```
int Actisite[4];
int Addr[4];
```

```
//Actisite[X]==0  为 ACTIVE SITE
Addr[1]=GET_DATA(6);//获取当前 SITE 信息

Actisite[0]=Addr[1]&0x01;//SITE1 ,1
Actisite[1]=Addr[1]&0x02;//SITE2 ,2
Actisite[2]=Addr[1]&0x04;//SITE3 ,4
Actisite[3]=Addr[1]&0x08;//SITE4 ,8

//VDSS
  SET_HVIS(FIMV,2,UA,800,V);    //FIMV
  Delay(50);

if(Actisite[0]==0)
{
      SET_RELAY("45");
         Delay(50);
      MeasureV[0] =MEASURE_HVIS();
}

if(Actisite[1]==0)
{
      SET_RELAY("46");
         Delay(10);
      MeasureV[1] =MEASURE_HVIS();
}

if(Actisite[2]==0)
{
      SET_RELAY("47");
         Delay(10);
      MeasureV[2] =MEASURE_HVIS();
}

if(Actisite[3]==0)
{
      SET_RELAY("48");
         Delay(10);
      MeasureV[3] =MEASURE_HVIS();
}



      CLEAR_HVIS();
      CLEAR_ALL();
```

```
 if(!SHOW_RESULT("VDSS",MeasureV,"V",850,300,2))
      return;


//IDSS
      SET_HVIS(FVMI,200,V,10,UA);
      Delay(10);
      SET_HVIS(FVMI,300,V,10,UA);
      Delay(10);
      SET_HVIS(FVMI,400,V,10,UA);
      Delay(10);
      SET_HVIS(FVMI,500,V,10,UA);
      Delay(10);

 if(Actisite[0]==0)
{
      SET_RELAY("45");////
      Delay(5);
      MeasureI[0] =MEASURE_HVIS()*1e6;
}
   if(Actisite[1]==0)
{
      SET_RELAY("46");////
      Delay(5);
      MeasureI[1] =MEASURE_HVIS()*1e6;
}
 if(Actisite[2]==0)
{
      SET_RELAY("47");////
      Delay(5);
      MeasureI[2] =MEASURE_HVIS()*1e6;
}
 if(Actisite[3]==0)
{
      SET_RELAY("48");
      Delay(5);
      MeasureI[3] =MEASURE_HVIS()*1e6;
}
 if(!SHOW_RESULT("IDSS",MeasureI,"uA",2,-2,2))
      return;
```