

目 录

第一章 简介	3
1.1 特性	4
1.2 标准配备	4
1.3 选用配备	4
1.4 规格	5
1.5 系统方块图	6
第二章 安装	7
2.1 安装前的准备	7
2.2 电源的设定与检查	7
2.3 接地需求	8
2.4 脚架调整	8
2.5 仪器箱的装设	8
2.6 环境需求	8
2.7 维修及校正服务	8
2.8 GPIB 介面功能	8
2.9 RS.232C 介面功能	9
2.10 遥控装置	9
第三章 操作说明	10
3.1 支撑脚	10
3.2 电源开关	10
3.3 开机状态	10
3.4 储存/呼叫 (STORE/RECALL) 操作	11
3.5 AUTO SEQUENCE测试功能说明	11
第四章 GPIB /RS.232 操作命令说明	13
4.1 GPIB/RS.232C 简介	13
4.2 GPIB 命令摘要	13
4.3 RS.232C 命令摘要	13
4.4 3250/3310A/3310A/3310C GPIB/RS232C命令列表	15
4.5 缩写代号说明	21
4.6 GPIB /RS.232C 命令说明	22
附录1 3310A 系列 GPIB 程式范例	37
附录2 3310A 系列 RS.232 程式范例	41
附录3 3250 系列 GPIB/RS.232C 操作流程	55
附录4 3310A 系列 GPIB/RS.232C 操作流程	56

图形

图 1.1	3300C 系统方块图	6
图 2.1	电源设定图	7
图 2.2	保险丝座	7
图 2.3	后面板	8
图 2.4	遥控连接埠图	9
图 2.5	类比驱动信号输入	9
图 3.1	3300C 前面板图	10
图 4.1	后面板 RS-232C 介面连接图	14

表格

表 1.1	3250/3310A/3310C/3330A 系列简单规格表	3
表 1.2	3300C 规格表	5
表 4.1	命令结束字元表	21
表 4.2	波形资料表	22
表 4.3	3250 系列 ERR 状态暂存器	31

第一章、简介

Model 3300C 电子负载机框乃是为了 3250/3310A/3310C/3330A 系列抽取式电子负载模组而设计的 4 组控制机框，3300C 可同时安装 4 组电子负载模组，当安装 3330A 系列模组时更可同时使用多达 8 组的电子负载，除了满足 4 组输出的电源供应器测试，更可满足 8 组输出的电源供应器测试。请注意3250/3310A/3310C/3330A 系列可同时安装于 3300C 机框上，目前本公司产生的 3250/3310A/3310C/3330A 系列抽取式电子负载模组的简单规格如表1.1所示，详细规格请参考各系列电子负载使用手册，或与本公司营业部联络取得详细资料。

型 号	最大电流	最大电压	最大功率
3250	20 Arms	60 Vrms	300 W
3251	8 Arms	150 Vrms	300 W
3252	4 Arms	300 Vrms	300 W
3253	1Arms	300 Vrms	300 W
3310C	30A	60V	150W
3311C	60A	60V	300W
3312C	10A	250V	300W
3314C	5A	500V	200W
3315C	15A	60V	75W
3310A	30 A	60 V	150 W
3311A	60 A	60 V	300 W
3312A	10 A	250 V	300 W
3314A	5 A	500 V	200 W
3315A	15 A	60 V	75 W
3330A	50A/5A	+60V/+60V	250W/50W
3331A	50A/5A	+60V/-60V	250W/50W
3332A	5A/5A	+60V/+60V	75W/75W
3333A	5A/5A	+60V/-60V	75W/75W
3334A	5A/5A	-60V/-60V	75W/75W
3335A	100A/5A	+60V/+60V	500W/50W

表 1.1 3250/3310A/3310C/3330A系列简单规格表

1.1 特性

Model 3300C 是一个操作简单、价格经济型的 4 组电子负载机框，包含下列特性：

- 1 电子负载模组采用可抽取式设计，方便更换不同规格之电子负载模组，并且能识模组的型号。
- 2 可任意使用组合安装 3250, 3310A, 3330C, 3330A 系列抽取是电子负载模组。
- 3 对于 3250, 3310A 系列电子负载模组, 3300C 可储存/呼叫 5 种电子负载状态设定项目；对于 3310C, 3330A 系列电子负载模组, 3300C 可储存/呼叫 5 种电子负载状态设定项目，每个项目可分为 30 种状态，共有 150 种电子负载状态设定。
- 4 具有RS-232C 界面功能，可连接 PC, NOTE BOOK PC 等作自动测试。
- 5 选用配备 GPIB 界面卡，可轻易达成自动化控制。
- 6 GPIB 位址由前板 STATE 4+STATE 5 同时按下进入 GPIB 位址设定模式，显示出 GPIB 位址按 STATE 2 跳出 GPIB 位址设定模式。
- 7 对机箱内部散热风扇采用智慧型控制，节约能源。
- 8 配备 9931C 遥控器，提供使用者远端操作 5 组或 150组 呼叫功能，用“NG”指示灯显示当前的测量状态，最适合线上使用。
- 9 3300C可以两台并联，使用一个 9931C 遥控器控制。

1.2 标准配备

- | | |
|----------------------|-----|
| 1 Model 3300C 主机 | 1 台 |
| 2 BNC-BNC 电缆线，长度 1 米 | 1 条 |
| 3 Model 3300C 使用手册 | 1 本 |
| 4 三端式电源线 | 1 条 |

1.3 选用配备

- 1 IEEE-488 介面。
- 2 GPIB 缆线长度 1 米。
- 3 GPIB 缆线长度 2 米。
- 4 空白面板。
- 5 D-SUB 9 Pin to D-SUB 9 Pin 连接电缆线长度 1 米。(9931 用)
- 6 9931C 遥控器。

1.4 规格

AC INPUT	LINE	115V ± 10%	230V ± 10%
	FREQUENCY	50/60 Hz	
	FUSE	2A/250V(5520mm)	1A/250V(5520mm)
	MAX.POWER CONSUMPTION	100W	
DIMENSIONS (W*H*D)		440mm5177mm5445mm	
WEIGHT		NET:9.3Kg	

表1.2 3300C 规格表

1.5 系统方块图

3300C 系统方块图如图1.1所示，3300C 内含 5 组电源供应器，其中一组供应给 3300C 机框，其余 4 组分别供应给电子负载模组。

3300C 电子负载机框与每一组电子负载模组以光隔离元件隔离。

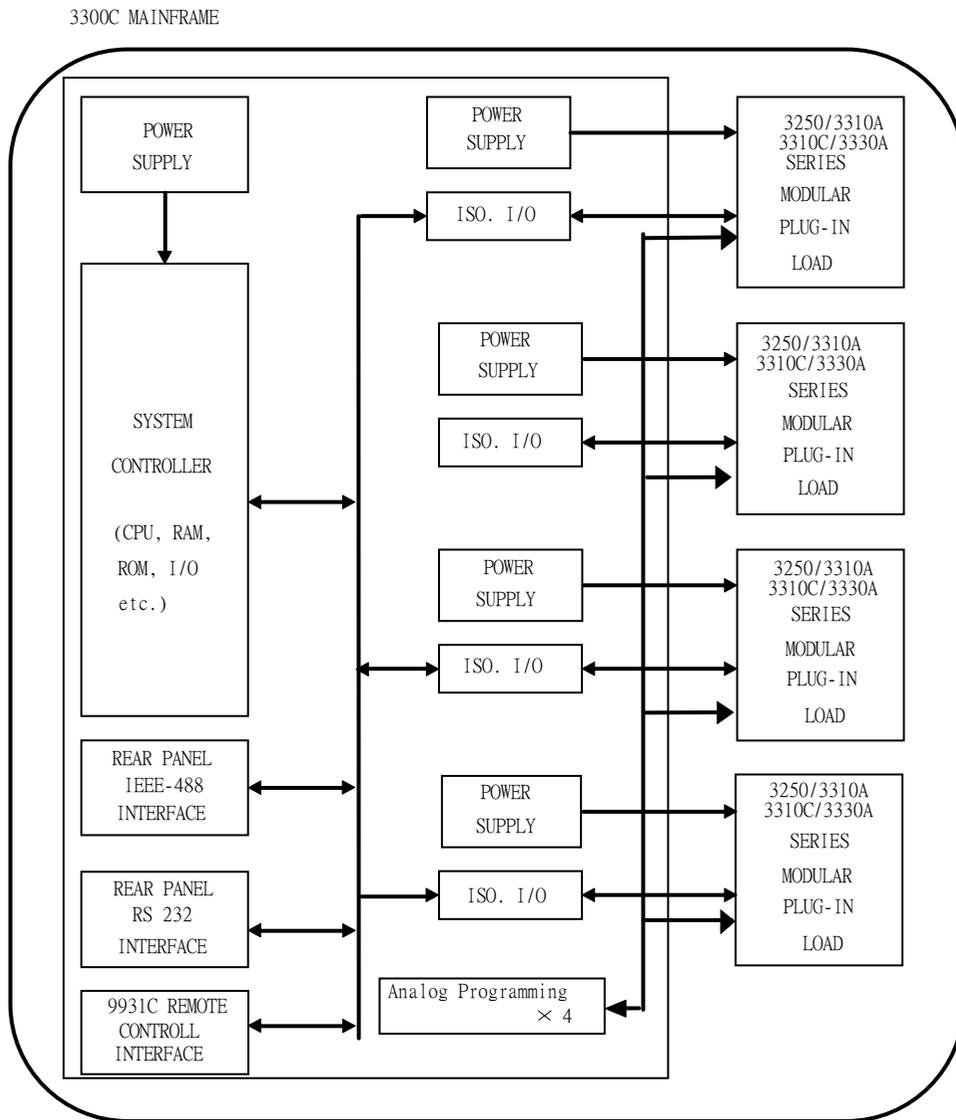


图1.1 3300C系统方块图

第二章、安装

2.1 安装前的准备

3300C 电子负载机框于出货前都已经过严密的品质检验，如果机框于运输过程遭受损坏时，请您就近联络博计电子的经销商或直接与本公司营业部联络。

2.2 电源的设定与检查

3300C 电子负载机框可以工作于交流电源 100/115V 及 200/230V，工作电压标示于后面板电源输入端附近，使用前请先确定标示的工作电压与您的使用电压是否相同，如果您的使用电压与 3300C 电子负载机框所标示的工作电压不同时，请依照以下的步骤重新设定工作电压。

2.2.1 关闭 3300C 电子负载机框前面板之电源开关 (0 的位置)。

2.2.2 设定开关位于机框后面板上，请参考图2.1设定正确的工作电压，电压的设定说明如下：

a 设定开关到 115V 位置即设定使用电压为 100/115 V。

B 设定开关到 230V 位置即设定使用电压为 200/230 V。

※ 100 及 200 伏特为日本之电源规格，仅提供给日本地区使用。

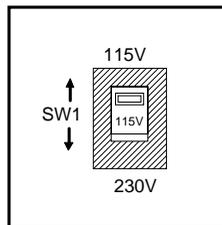


图2.1 电源设定图

2.2.3 确认保险丝的安装是否正确，如果必要时，请一并更换正确的保险丝，一般来说应该为另一颗位于保险丝座中的备用保险丝。

2.2.4 保险丝座位于交流电源插座下方，检视保险丝前务必先拔除电源线，以避免电击的危险，取出保险丝座时，图2.2所示可以使用一把较小的平头螺丝起子，换上如表1.2所示正确规格的保险丝。

2.2.5 置回保险丝座，插上电源线后即可。

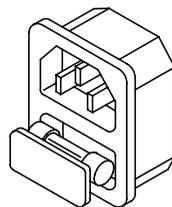


图2.2 保险丝座

2.3 接地需求

为了避免机壳因漏电时而造成危险，3300C 电子负载机框强烈要求使用三端式的电源线，并且电源配线接地皆需正确和完整。

2.4 脚架调整

3300C 电子负载机框为桌上型测试设备，脚架的使用可以提供更佳的观测点，使用时仅须将脚架向外弹出即可。

2.5 仪器箱的装设

3300C 除了可以置放于工作台上之外，更可以固定于标准 19 英寸的仪器箱上使用。使用时直接将 3300C 置放于仪器箱中，将左右把手固定（锁）于仪器箱上即可。

2.6 环境需求

为了保持 3300C 电子负载机框可以正常地操作，建议操作环境的温度应位于摄氏 0℃ ~ 40℃ 之间，最佳的工作环境温度为摄氏 25℃ ± 5℃。

2.7 维修及校正服务

如果 3300C 电子负载机框故障或需要校正时，请于机框上贴上标示有所有人(公司行号部门人员)的标签，并指明为校正服务或者维修服务，然后通知博计电子的经销商或者直接与本公司联络。

2.8 GPIB 介面功能

GPIB 连接器位于 3300C 电子负载机框的后面板，用于连接 GPIB 控制器 (CONTROLLER) 或其他装置 (DEVICES)。

GPIB 连线时有二点较为重要的限制如后所述：

2.8.1 包含 GPIB 控制器 (CONTROLLER) 在内，所有装置不能超过 15 台。

2.8.2 GPIB 连接器电缆线长度最长为 2 米，装置连线后其总长不可超出 20 米。

2.8.3 图 2.3 显示 3300C 电子负载机框的后面板，GPIB 连接头位于 3300C 后面板上，GPIB 的地址在 3300C 前面板设定。

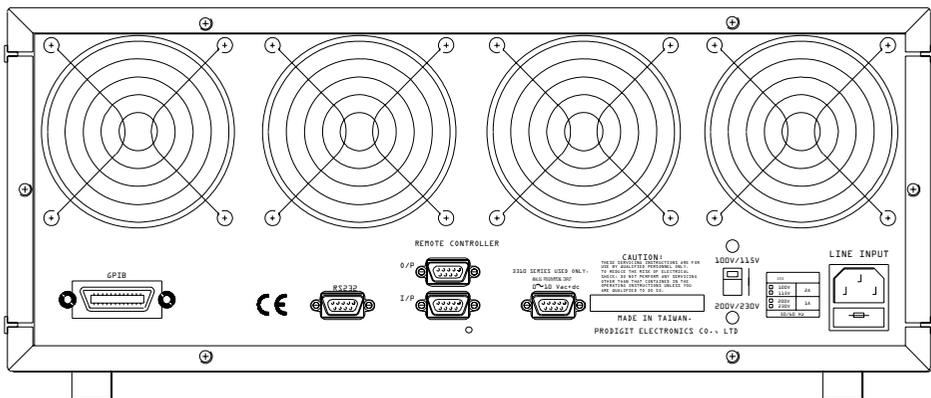


图 2.3 后面板

2.9 RS-232C 介面功能

3300C 电子负载机框提供了一个 RS-232C 母座 (FEMALE) 连接装置于背面板上，此连接装置与电脑 RS-232C 连接埠以一对一的方式连接。

2.10 遥控装置

3300C 电子负载机框提供了二个可以连接 Model 9931C 遥控器的连接埠于后面板上，I/P及O/P连接埠为一般的 9 Pin 母座连接埠，I/P埠与前面板的按键连接，用以使用 Model 9931C 遥控器，取代 3300C 前面板的 Recall 和“UP”和“DOWN”按键如果 3300C 机框中有任一电子负载产生 NG，MODEL 9931C 上的“NG”指示灯指示 NG 状态。

遥控输出端口能连接到其它的3300C、3300A或3302机框，该功能使用户可同时对12个或更多的模组进行存储/呼叫操作，尤其适合测试多输出电源的场合。

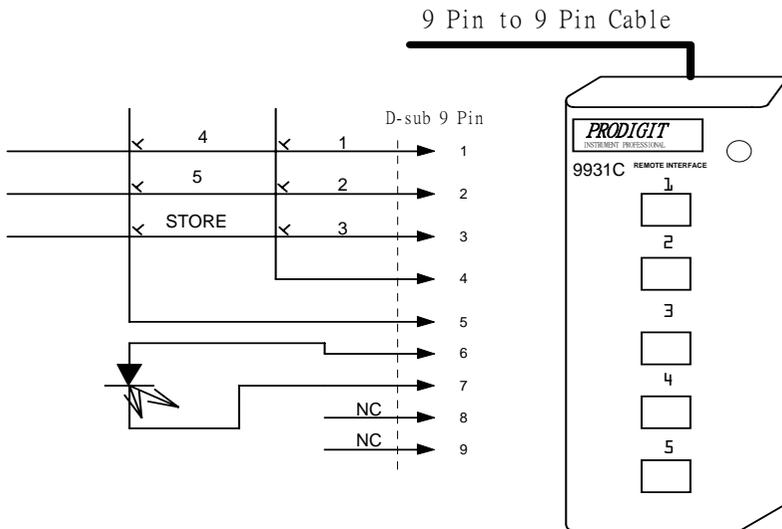


图2.4 遥控连接埠图

2.11 类比驱动信号输入

3300C机框后面板的D-Sub 9针接口为3310A/3310C系列模组提供类比驱动信号或为3250系列模组提供外部SYNC信号。接口定义见图2.5。

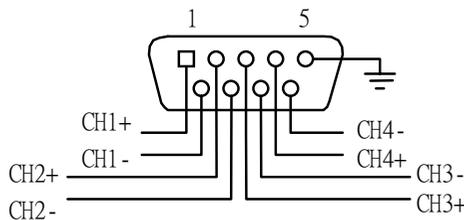


图2.5 类比驱动信号输入

第三章、操作说明

3300C 电子负载机框前面板图，如图3.1所示。

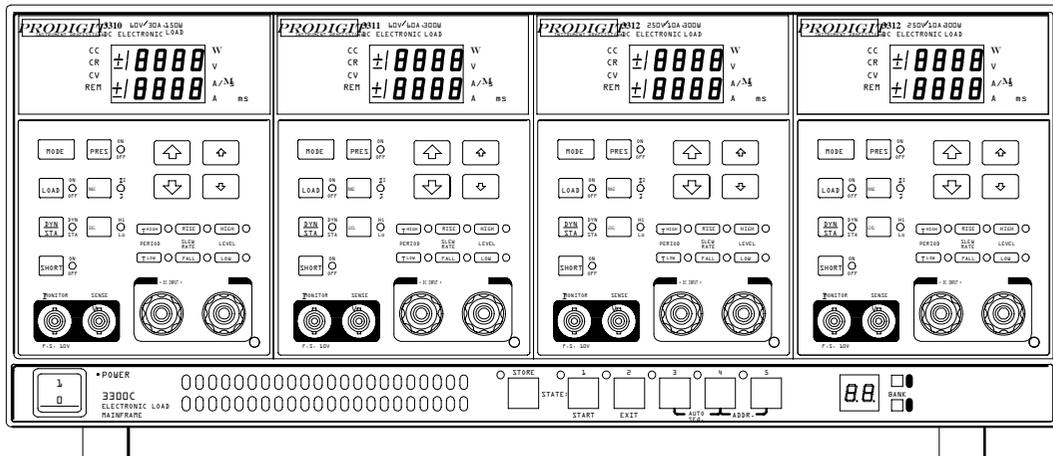


图3.1 3300C 前面板图

3.1 支撑脚

在3300C底板装有4个支撑脚，接近3300C前面板的2个支撑脚可竖立，以便操作者得到更好的观察角度。

3.2 电源开关

在接上电源线之前，请先确定使用电压与本机所使用的电压是否相符。

当电源开关切换到"1"(ON) 时，3300C电子负载机框及 3250/3310A/3310C/3330A系列电子负载模组即能正常地工作。

3.3 开机状态

3300C:

3.3.1 手动 (面板) 操作状态。

3.3.2 储存/呼叫: 所有LED指示器熄灭, BANK数码管显示“01”。

3.3.3 3250/3310A/3310C/3330A 系列电子负载处于初始状态, 详情请参阅电子负载模组的操作手册。

3.3.4 GPIB地址设定: 同时按下“STATE 4”和“STATE 5”2个按键, 进入GPIB地址设定模式, 按“UP”和“DOWN”键选择0~31的地址码, 按“STATE 2”退出GPIB地址设定模式。

3.4 储存/呼叫 (STORE/RECALL) 操作

3300C 电子负载机框前面板的 8 个功能键, 对于 3250、3310A 系列电子负载模组其中 STATE1~5 提供了使用者可以储存最多达 5 个的测试项目或状态, 对于 3310C、3330A 系列电子负载模组 3300C 可储存/呼叫 5 种电子负载状态设定项目, 每个项目别分为 30 种状态, 共有 150 种, 电子负载状态设定于 3300C 中的 EEPROM 中, 每一组 STATE(1-5) 皆能储存(STORE)或呼叫(RECALL) 3250/3310A/3310C/3330A 电子负载的状态及设定值。

储存功能操作步骤:

- 3.4.1 设定好电子负载的状态及设定值。
- 3.4.2 对于 3310C 和 3330A 系列, 用 UP 和 DOWN 键选择将要存储的 BANK 状态号码 (1~30), 对于 3310A 和 3250A 系列, 跳到下一步。
- 3.4.3 按下 3300C 面板上的储存(STORE)键, 此时储存件的指示灯会立即以每秒一次的速度闪烁。若欲放弃储存时, 可再按一次储存键或等大约 20 秒后即离开储存功能。
- 3.4.4 按下储存键后, 储存功能指示灯亦开始闪烁之后, 按下 STAT1~5 任何一键时, 相对地指示灯立即点亮。表示电子负载面板状态及设定值都已经储存至指定的记忆装置中。储存功能指示灯熄灭之后, 表示储存步骤已经完成。

呼叫功能操作步骤:

按下 STATE 1~5 中任何一个按键, 相对的指示灯即点亮, 表示 3250, 3310A 系列电子负载模组, 会从相对的记忆装置中将资料呼叫出来, 此时 3250, 3310A 系列电子负载模组面板的状态及设定值即会依照呼叫出来的资料重新设定, 对于 3310C 和 3330A 系列, 按 UP 和 DOWN 键改变当前项目中的状态号码, 同时将此状态的资料传送至电子负载模组。设定呼叫功能后, 若按下模组面板上任何一个按键时, 呼叫功能指示灯 (STATE 1~5) 随即熄灭, 表示呼叫步骤已经完成。

3.5 AUTO SEQUENCE 测试功能说明

在 AUTO SEQUENCE 功能中有两种工作模式: 编辑模式和测试模式。按下 S3+S4 键进入 AUTO SEQUENCE 状态, 然后按 STORE 进入编辑模式, 或按 STATE 进入测试方式, 请参考下方的流程图。

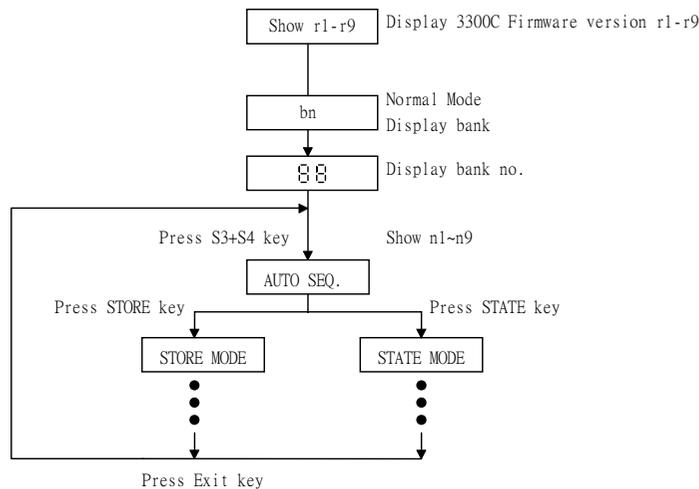


图 3.3 AUTO SEQUENCE 功能操作流程

3.5.1 编辑状态

进入AUTO SEQ状态后，S3和S4的LED会同时点亮，此时如按下STORE键即进入编辑模式。

3.5.1.1 3300C最多能编辑9个AUTO SEQ。

3.5.1.2 每个AUTO SEQ最多可设置16个测试步骤。每个测试步骤可以是150组记忆状态中的任一记忆状态。

3.5.1.3 每个测试步骤含 t_1 （测试时间）和 t_2 （延迟时间），以100mS为基本单位。 t_1/t_2 设定范围为0.1Sec~9.9Sec，每个测试步骤开始后经过 t_1 （测试时间），才会进行“GO/NG”的判别。经过 t_2 （延迟时间）之后，下一步骤开始。

3.5.1.4 AUTO SEQ最多可设置16个测试步骤，设置完成后，按EXIT键退出AUTO SEQ编辑模式。

3.5.2 测试状态

3.5.2.1 AUTO SEQ状态下，S3和S4的LED同时亮起时，按START键进入测试模式。

3.5.2.2 顺序启动（步骤0- t_1-t_2 ），然后（步骤1- t_1-t_2 ），如此直到执行完所有预先设定步骤。如过程中按EXIT键，将会停止测试并退出测试模式。

3.5.2.3 如果3300C内所有的模组全部测试步骤均PASS，LED数码管显示GO（闪烁），如果任何一个模组在测试中FAIL，则3300C会停止测试，并显示nG（闪烁）。

3.5.2.4 用户可按START键继续其它的测试，或按EXIT退出测试模式及AUTO SEQ状态。

第四章、GPIB /RS-232 操作命令说明

4.1 GPIB/RS-232C 简介

3300C 电子负载机框后面板上的 GPIB /RS-232C 介面可以和个人电脑 (PC) 或者笔记本型电脑 (Note Book PC) 的 GPIB /RS-232C 介面连接, 可以使用高阶语言 C 和 VB 等应用程式, 远端控制电子负载模组, 组成自动控制系统。

根据 GPIB /RS-232C 介面功能, 更可以利用在交换式电源供应器 (Switching Mode Power Supply) 的自动化测试, 如负载调整率 (LOAD Regulation), 电压调整 (Centering Voltage Adjust) 等, 或者可充电式电池的充放电测试。3300C 的 GPIB /RS-232C 介面功能, 不仅可以设定 3250.3310A. 3330A.3310C 系列电子负载模组的负载状态, 更可以读回设定值及实际值, 从而可以在 PC 上可以观察到电子负载模组的工作状态。

4.2 GPIB 命令摘要

GPIB 的设定命令中除了 "SYSTEM, GLOBE" 之外, 其余命令都属于组别限定命令 (Channel Dependent Command), 即每个设定命令都是在指定组别中设定, 所以在应用程式设计时, 每一个设定命令之前必须先确定于那一个组别。

例:

欲设定负载 2 Sink 电流时, GPIB 命令下达时应如下所述:

```
CHAN 2; LOAD ON
```

若再欲设定负载 2 不 Sink 电流时, GPIB 命令下达时即可不再下达 CHAN 2 命令, 只须下达 LOAD OFF, 因为目前即位于组别 2 的环境中。

GPIB 命令中增加了 [GLOB:] 选项时, 表示所下达的命令所有组别都接受, 即所有组别同时接受同一命令的下达, 这项特点大大地减少了测试时间。

4.3 RS-232C 命令摘要

RS-232C 命令语法与 GPIB 命令语法都是相同的, 3300C 电子负载机框 RS-232C 功能的通讯协定为固定的, 其通讯协定如下所述。

鲍得率 (Baud-rate)	: 9600
同位检查 (Parity)	: 有
资料位元数 (Data bit)	: 8 位元
结束位元 (Stop bit)	: 1 位元
命令延迟时间 (Command Delay time)	: 20 mSec

后面板 RS-232C 介面连接图如图 4-1, 其中图 4-1(A) 为 3300C RS-232C 介面的内部配线图。使用者只须使用如图 4-1(B) 的一般一对一 RS232-C 电缆线。

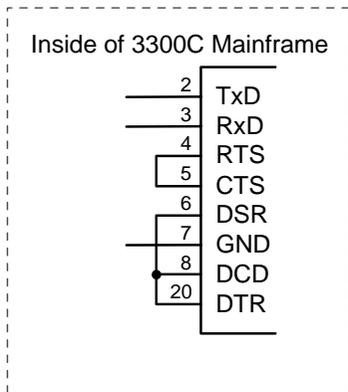


Figure 4-1.A

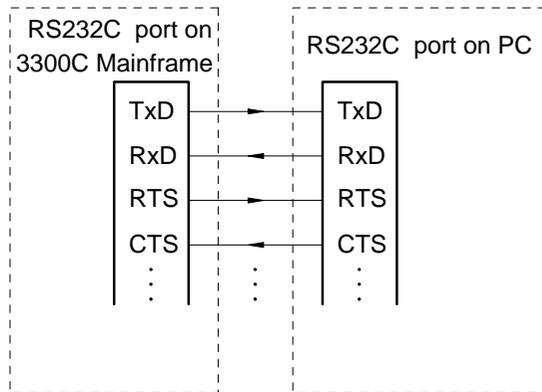


Figure 4-1.B

图4.1 后面板 RS-232C 介面连接图

4.4 3250/3310A/3310C/3330A GPIB/RS-232C 命令列表

3250/3310A/3310C/3330A 系列 GPIB/RS-232C 设定及读取命令如下表。

设定预置数值命令	型 号				备注註
	3310A	3310C	3330A	3250	
[PRESet:] BANK{SP}{d};; NL}					d=0~10
[PRESet:] WAVE{SP}{m};; NL}					M=0~5
[PRESet:] PREQuency {SP} {NR2} ;; NL}					0.1~999Hz
[PRESet:] RISE{SP} {NR2} ;; NL}					
[PRESet:] FALL{SP} {NR2} ;; NL}					
[PRESet:] SLEWrate{SP} {NR2} ;; NL}					
[PRESet:] PREIod: HIGH LOW {SP} {NR2} ;; NL}					
[PRESet:] LDONv{SP} {NR2} ;; NL}					
[PRESet:] LDOFv{SP} {NR2} ;; NL}					
[PRESet:] CC{SP} {NR2} ;; NL}					
[PRESet:] CC: {A B} {SP} {NR2};; NL}					
[PRESet:] CC: {HIGH LOW} {SP} {NR2};; NL}					
[PRESet:] CP: {HIGH LOW} {SP} {NR2};; NL}					
[PRESet:] CR: {SP} {NR2} ;; NL}					
[PRESet:] CR: {A B} {SP} {NR2};; NL}					
[PRESet:] CR: {HIGH LOW} {SP} {NR2};; NL}					
[PRESet:] CV: {HIGH LOW} {SP} {NR2};; NL}					
[PRESet:] CV: {SP} {NR2} ;; NL}					
[PRESet:] OCP: START{SP} {NR2} ;; NL}					
[PRESet:] OCP: STEP{SP} {NR2} ;; NL}					
[PRESet:] OCP: STOP{SP} {NR2} ;; NL}					
[PRESet:] OCP: Vth{SP} {NR2} ;; NL}					

查询预置数值命令	型 号				RETURN
	3310A	3310C	3330A	3250	
[PRESet:] BANK{SP}{?}; NL}					0~10
[PRESet:] WAVE{SP}{?}; NL }					1~5
[PRESet:] FREQuency {?}; NL}					0.1~999
[PRESet:] RISE {?} ; NL}					####.#####
[PRESet:] FALL{?} {NR2} ; NL}					####.#####
[PRESet:] SLEWrate {?} ; NL}					####.#####
[PRESet:] PREIod: {HIGH LOW}{?} ; NL}					####.#####
[PRESet:] LDONv {?}; NL}					####.#####
[PRESet:] LDOFfv {?}; NL}					####.#####
[PRESet:] CC{?} ; NL}					####.#####
[PRESet:] CC: {A B} {?}; NL}					####.#####
[PRESet:] CC: {HIGH LOW} {?} ; NL}					####.#####
[PRESet:] CP: {HIGH LOW} {?} ; NL}					####.#####
[PRESet:] CR: {?} ; NL}					####.#####
[PRESet:] CR: {A B} {?}; NL}					####.#####
[PRESet:] CR: {HIGH LOW} {?} ; NL}					####.#####
[PRESet:] CV: {HIGH LOW} {?} ; NL}					####.#####
[PRESet:] CV: {?} ; NL}					####.#####
[PRESet:] OCP: START{?} ; NL}					####.#####
[PRESet:] OCP: STEP{?} ; NL}					####.#####
[PRESet:] OCP: STOP{?} ; NL}					####.#####
[PRESet:] OCP: Vth{?} ; NL}					####.#####

LIMIT命令	型 号				RETURN
	3310A	3310C	3330A	3250	
LIMIT: CURRent: {HIGH LOW}{SP}{NR2}; NL}					
LIMIT: CURRent: {HIGH LOW}{?}; NL}					####.#####
LIMIT: POWer: {HIGH LOW}{SP}{NR2}; NL}					
LIMIT: POWer: {HIGH LOW}{?}; NL}					####.#####
LIMIT: VA: {HIGH LOW}{SP}{NR2}; NL}					
LIMIT: VA: {HIGH LOW}{?}; NL}					####.#####

LIMIT: VOLTage: {HIGH LOW}{SP}{NR2}{;} NL}					
LIMIT: VOLTage: {HIGH LOW}{?}{;} NL}					###.####

STAGE命令	型 号				RETURN
	3310A	3310C	3330A	3250	
[STATe:]LOAD {SP}{ON OFF} {;} NL}					
[STATe:]LOAD {?} {;} NL}					0: OFF 1: ON
[STATe:]MODE {SP}{CC CR CV CP} {;} NL}					
[STATe:]MODE {?} {;} NL}					0: CC 1: CF 2: CV 3: CF
[STATe:]SHORt {SP}{ON OFF} {;} NL}					
[STATe:]SHORt {?} {;} NL}					0: OFF 1: C
[STATe:]PRESet {SP}{ON OFF} {;} NL}					
[STATe:]SENSet {?} {;} NL}					0: OFF 1: C
[STATe:]SENSe {SP}{ON OFF} {;} NL}					
[STATe:]PRESe {?} {;} NL}					0: OFF 1: C
[STATe:]RANGe {SP}{I II} {;} NL}					
[STATe:]RANGe {?} {;} NL}					0: OFF 1: C
[STATe:]RANGe {SP}{I II} {;} NL}					
[STATe:]RANGe {?} {;} NL}					0: I 1: II
[STATe:]LEVE1 {SP}{HIGH LOW} {;} NL}					
[STATe:]LEVE1 {?} {;} NL}					0: LOW 1:
[STATe:]DYNamic {SP}{ON OFF} {;} NL}					
[STATe:]DYNamic {?} {;} NL}					0: OFF 1: C
[STATe:]SYNCronize {SP}{ON OFF} {;} NL}					
[STATe:]SYNCronize {?} {;} NL}					0: OFF 1: C
[STATe:]WATT {SP}{ON OFF}					
[STATe:]WATT {?} }					0: OFF 1: C
[STATe:]CLRerr					
[STATe:]ERRor {?} }					0: 1: ERF
[STATe:]DUAL{SP}{ON OFF}					
[STATe:]NG {?} }					0: 1NG
[STATe:]PROTect {?} }					0: 1: PROT
[STATe:]LIMit{? } }					
[STATe:]NGABle[SP]{ON OFF}					

[STATe:]NGABLE{? }					
[STATe:]PARAllel[SP]{ON OFF}					
[STATe:]SAVEcal					
[STATe:]CLRTest					
[STATe:]TCONfig[SP]{NORM OCP OVP}					
[STATe:]TCONfig{? }					
[STATe:]TMODE{? }					
[STATe:]TSTEP[SP]{FAST NORM SLOW}					
[STATe:]TSTEP{? }					

系统命令：对所有模组有效

COMMAND	Note	RETURN
[SYStem:]INITial{;} NL}	Only 3330A	
[SYStem:]CHANnel{SP}{1 2 3 4}[sp][,][A B]{;} NL}		
[SYStem:]CHANnel{SP}{?}{;} NL}		1 2 3 4[A B]
[SYStem:]RECall{sp}{m[,n]}{;} NL}	M=1~5 n=1~30	
[SYStem:]STORe{sp}{m[,n]}{;} NL}	M=1~5 n=1~30	
[SYStem:]REMOtE{;} NL}	Only RS232 cmd	
[SYStem:]LOCAL{;} NL}	Only RS232 cmd	0:OFF 1:ON
[SYStem:]NAME{?}{;} NL}		"XXXXXX"

测量命令：对所有模组有效

COMMAND	3310A	3310C	3330A	3250	RETURN
MEASure: CURRent {?}{;} NL}					###.####
MEASure: VOLtage {?}{;} NL}					###.####
MEASure: POWER{?}{;} NL}					###.####
MEASure: OCP {?}{;} NL}					###.####
MEASure: OVP {?}{;} NL}					###.####

调校命令	型 号		
	3310A	3310C	3330A
{OFFSet GAN}: CC: I: {AC DC}{SP}{NR2}{;} NL}			
{OFFSet GAN}: CC: I: {HIGH LOW}{SP}{NR2}{;} NL}			
{OFFSet GAN}: CC: II{AC DC}{SP}{NR2}{;} NL}			
{OFFSet GAN}: CC: II: {HIGH LOW}{SP}{NR2}{;} NL}			
{OFFSet GAN}: CP: I: {HIGH LOW}{SP}{NR2}{;} NL}			
{OFFSet GAN}: CR: I{SP}{NR2}{;} NL}			
{OFFSet GAN}: CR: I: {AC DC}{SP}{NR2}{;} NL}			
{OFFSet GAN}: CR: I: {HIGH LOW}{SP}{NR2}{;} NL}			
{OFFSet GAN}: CR: II: {SP}{NR2}{;} NL}			
{OFFSet GAN}: CR: II: {AC DC}{SP}{NR2}{;} NL}			
{OFFSet GAN}: CR: II: {HIGH LOW}{SP}{SR2}{;} NL}			
{OFFSet GAN}: CV: I{SP}{NR2}{;} NL}			
{OFFSet GAN}: CV: I: {HIGH LOW}{SP}{SR2}{;} NL}			
{OFFSet GAN}: CV: II: {SP}{NR2}{;} NL}			
{OFFSet GAN} : MEASure : CURRent : {AC DC}{SP}{NR2}{;} NL}			
{OFFSet GAN} : MEASure : CURRent : {I II}{SP}{NR2}{;} NL}			
{OFFSet GAN} : MEASure : POWer: {AC DC}{SP}{NR2}{;} NL}			
{OFFSet GAN} : MEASure : VOLtage : {AC DC}{SP}{NR2}{;} NL}			
{OFFSet GAN} : MEASure : VOLtage : {I II}{SP}{NR2}{;} NL}			
{OFFSet GAN} : MEASure : VOLtage : SENSE : {I II}{SP}{NR2}{;} NL}			

GLOBE命令

COMMAND	3310A	3310C	3330A	3250	RETU
GLOBe: [STATe:] PRESet {SP}{ON OFF};; NL}					
GLOBe: [STATe:] LOAD {SP}{ON OFF};; NL}					
GLOBe: [STATe:] MODE {SP}{ON OFF};; NL}					
GLOBe: [STATe:] SHORt {SP}{ON OFF};; NL}					
GLOBe: [STATe:] DYNamic {SP}{ON OFF};; NL}					
GLOBe: [STATe:] LEVEL {SP}{HIGH LOW};; NL}					
GLOBe: [STATe:] LEVEL {SP}{I II};; NL}					
GLOBe: [STATe:] RANGe {SP}{I II};; NL}					
GLOBe: MEASure: CURRent{?};; NL}					###.# #
GLOBe: MEASure: VOLtage{?};; NL}					###.# #

Auto SEQUENCE: DOWNload[SP] n {SP|,} NR3

AUTO: DOWNload [SP] n {SP ,}NR3	8>n>1,NR 3: REC. T1H. T1L. T2H. T2L. SP
AUTO: Upload [SP] n?	8>n>1, NR 3*16
AUTO: SEQUENCE n	8>n>1, return 0: AUTO SEQUENCE 1-15; AUTO SEQ
AUTO: NG?	1: NG

附:

1. 电流单位为安培 (A/Arms)
2. 电阻单位为欧姆 (Ω)
3. 电压单位为伏特 (V/Vrms)
4. 周期单位为毫秒 (mS)
5. 转换率 (SLEW-RATE) 单位为安培/微秒 (A/uS)
6. 频率单位为赫兹 (Hz)
7. 功率单位为瓦特 (W)
8. VA 单位为伏安 (VA)

4.5 缩写代号说明

- 1 SP: SPACE, 空隔字元, ASCII 码为 20H.
- 2 ; : 命令结束符号。
- 3 NL: 命令结束符号。
- 4 NR2: 包含小数点的数值形式, 形式为 #.#.### 在此范围内皆可接受。
例如: 30.1234, 5.0

GPIB / RS-232C 命令语法说明

- 1 { } : 此符号表示命令必需包含此项, 不可省略。
- 2 [] : 此符号表示命令中可以有, 可以没有此项参数。
- 3 | : 此符号表示 OPTION 之意, 例如: "LOW|HIGH"表示可以使用 LOW 或 HIGH, 但两者只能选择其中一个使用。
- 5 在下达完一个命令后, 你必须接者送出一个命令结束字元, 本机可接受之结束字元为如表 4.1, 或同时送出多个命令, 每个命令之间以分隔符号";"隔开在最后一个命令加上结束位元。若你未送出结束字元, 则此命令视为无效命令。

LF
LF WITH EOI
CR, LF
CR, LF WITH EOI

表4.1 命令结束字元表

- 6 当一开始下达命令后, 你会看到电子负载上 REM 指示灯会亮起, 表示已经进入 Remote 状态。当你欲结束 GPIB 控制时, 请下达 GTL (GOTO LOCAL) 命令, 电子负载才会恢复 LOCAL 状态 (REM 指示灯熄灭)。

4.6 GPIB /RS-232C 命令说明

4.6.1 PRESET 设定和读取电子负载的预设值

BANK (适用机型: 3250 系列)

格式: [PRESet:]BANK{SP}{d}{; |NL} d=0~10
 [PRESet:]BANK? {; |NL}

用途: 设定和读取所选择的波形库

说明: 此命令为设定欲选择的波形库。

- 1) 波形库 0~4 为正弦波。
- 2) 波形库 5~9 为方波。
- 3) 波形库 10 为直流。
- 4) 每个波形库当中有 5 个波形资料, 共11个波形库, 55个波形资料, 波形资料如表4.2。

	波形库	A	B	C	D	E
正弦波	0	$\sqrt{2}$	2.0	2.5	3.0	3.5
	1	1.5	1.6	1.7	1.8	1.9
	2	2.0	2.1	2.2	2.3	2.4
	3	2.5	2.6	2.7	2.8	2.9
	4	3.0	3.1	3.2	3.3	3.4
方波	5	1.0	1.1	1.2	1.3	1.4
	6	1.5	1.6	1.7	1.8	1.9
	7	2.0	2.1	2.2	2.3	2.4
	8	2.5	2.6	2.7	2.8	2.9
	9	3.0	3.1	3.2	3.3	3.4
	10	$\sqrt{2}dc$	2dc	2.5dc	3.0dc	3.5dc

表4.2 波形资料表

WAVE (适用机型: 3250 系列)

格式: [PRESet:]WAVE{SP}{d}{; |NL} d=1~5
 [PRESet:]WAVE? {; |NL}

用途: 设定和读取所选择的波形库

说明: 此命令是设定 CC MODE 时的电流 C.F.(峰值因数)

此命令仅在 CC MODE 时才有作用, BANK 改变时这 5 组 C.F.亦同时会定义成不同之 C.F. 如表 4.4 详细内容请参考 3250 使用手册。

FREQ (适用机型: 3250 系列)

格式: [PRESet:] FREQQuery {SP}{NR2}{; |NL}
[PRESet:] FREQQuery ? {; |NL}

用途: 设定和读取频率值

说明: 此命令为设定电子负载的频率值, 下达命令时需注意下列事项:

- 1) 下达的频率值必须为含有小数点的数值, 否则命令无效。
数值最小有效位数为小数点后第 5 位。
- 2) 下达的数值若超过电子负载得规格时, 则 3302 机框会送出该电子负载规格的满刻度电流值。
- 3) 3250 系列电子负载的频率设为范围为 0.1 ~ 999 Hz。
- 4) 单位为赫兹 (Hz)

RISE (适用机型: 3310A, 3310C 系列)

格式: [PRESet:] RISE {SP}{NR2}{; |NL}
[PRESet:] RISE ? {; |NL}

用途: 设定和读取负载转换率 (SLEW-RATE) 的上升斜率

说明: 1) 负载转换率上升时间的定义包括负载电流改变时及动态负载电流 (DYNAMIC)。上升时间(RISE)与下降时间 (FALL) 的设定为完全独立。

- 2) 上升时间得设定值必须包含小数点的数值, 否则命令无效。
- 3) 数值最小有效位数为小数点后第 5 位。
- 4) 下达上升时间数值若超过电子负载得规格时, 则 3302 机框会送出该电子负载规格的满刻度上升时间。
- 5) 单位微安培/微秒 (A/uS)。

FALL (适用机型: 3310A, 3310C 系列)

格式: [PRESet:] FALL {SP}{NR2}{; |NL}
[PRESet:] FALL ? {; |NL}

用途: 设定和读取负载转换率 (SLEW-RATE) 的下降斜率

说明: 1) 负载转换率下降时间的定义包括负载电流改变时及动态负载电流。下降时间 (FALL) 与上升时间 (RISE) 的设定为完全独立。

- 2) 下降时间得设定值必须包含小数点的数值, 否则命令无效。
- 3) 数值最小有效位数为小数点后第 5 位。
- 4) 下达下降时间数值若超过电子负载得规格时, 则 3302 机框会送出该电子负载规格的满刻度电流值。
- 5) 单位微安培/微秒 (A/uS)。

SLEW (适用机型: 3330A 系列)

格式: [PRESet:] SLEWRATE{SP}{NR2}{; |NL}
[PRESet:] SLEW ? {; |NL}

用途: 设定和读取负载转换率 (SLEW-RATE) 的上升和下降斜率

说明: 请参考 RISE和 FALL

PERi (适用机型: 3310A, 3310C, 3330A 系列)

格式: [PRESet:] PERiod: HIGH|LOW{SP}{NR2}{; |NL}
[PRESet:] PERi? {; |NL}

用途: 设定和读取动态 (DYNAMIC) 负载时的 Tlow 和 Thigh宽度

说明: 1) 动态 (DYNAMIC) 负载波形的周期为 TLOW 与 THIGH的组成。

2) TLOW 与 THIGH 的的设定值必须为包含小数点的数值, 否则命令无效。

3) 数值的最小有效位数为小数点后第 5 位。

4) 下达的 TLOW 或 THIGH 数值超过电子负载的最大规格时, 3302 机框会送出该电子负载的满刻度 TLOW 或 THIGH 数值。

5) 单位为毫秒 (mS)

LDONv (适用机型: 3310A, 3310C, 3330A 系列)

格式: [PRESet:] LDONv {SP}{NR2}{; |NL}
[PRESet:] LDONv? {; |NL}

用途: 设定和读取LOAD ON 电压

说明: 此命令为设定电子负载 LOAD ON 电压值。

LDOFfv (适用机型: 3310C, 3330A 系列)

格式: [PRESet:] LDOFfv {SP}{NR2}{; |NL}
[PRESet:] LDOFfv ? {; |NL}

用途: 设定和读取负载 LOAD OFF 电压

说明: 此命令为设定电子负载 LOAD OFF 电压值。

CC (适用机型: 3330A 系列)

格式: [PRESet:] CC {SP}{NR2}{; |NL}
[PRESet:] CC ? {; |NL}

用途: 设定和读取负载电流值

说明: 设定电子负载欲载入的电流值,3330A 系列指 STATIC CURRENT

CC: HIGH|LOW (适用机型: 3310A, 3310C, 3330A 系列)

格式: [PRESet:] CC: HIGH|LOW{SP}{NR2}{; |NL}
[PRESet:] CC: HIGH|LOW ? {; |NL}

用途: 设定和读取负载 HIGH|LOW 组电流值

说明: 此命令为设定电子负载欲载入的电流值, 下达命令时须注意下列事项:

1) 下达的电流值必须含有小数点的数值, 否则命令无效。

2) 数值的最小有效位数为小数点后第 5 位。

3) 下达的电流数值超过该电子负载的最大规格时, 3302 机框会送出该电子负载规格的满刻度电流值。

4) LOW 的设定电流值必须比 HIGH 的设定电流值小; 如 5.3 定义说明。

5) 单位为安培 (A)。

CP: { HIGH|LOW} (适用机型: 系列 3310C 系列)

格式: [PRESet:] CP: { HIGH|LOW}{SP}{ NR2}{ ; |NL}

[PRESet:] CP: { HIGH|LOW} ? { ; |NL}

用途: 设定和读取负载固定功率值

说明: 此命令为设定电子负载欲载入的功率

CR (适用机型: 3330A 系列)

格式: [PRESet:] CR: {SP}{ NR2}{ ; |NL}

[PRESet:] CR? { ; |NL}

用途: 设定和读取负载电阻值

说明: 设定电子负载欲载入的静态电阻值

CR: { A|B} (适用机型: 3250系列)

格式: [PRESet:] CR: { A|B}{SP}{ NR2}{ ; |NL}

[PRESet:] CR: { A|B}? { ; |NL}

用途: 设定和读取负载电阻值

说明: 此命令为设定A组或B组电子负载欲载入的电流值, 下达命令时注意下列事项:

- 1) 下达的电流值必须含有小数点的数值, 否则命令无效。
- 2) 数值的最小有效位数为小数点后第 5 位。
- 3) 下达的电流数值超过该电子负载的最大规格时, 3302 机框会送出该电子负载规格的满刻度
电流值。
- 4) 单位为安培 (A)。

CR: { HIGH|LOW} (适用机型: 3310A, 3310C 系列)

格式: [PRESet:] CR: { HIGH|LOW}{SP}{ NR2}{ ; |NL}

[PRESet:] CR: { HIGH|LOW}? { ; |NL}

用途: 设定和读取负载电阻值

说明: 此命令为设定电子负载欲载入的电流值, 下达命令时注意下列事项:

- 1) 下达的电流值必须含有小数点的数值, 否则命令无效。
- 2) 数值的最小有效位数为小数点后第 5 位。
- 3) 下达的电流数值超过该电子负载的最大规格时, 3302 机框会送出该电子负载规格的满刻度
电流值。
- 4) LOW的设定电流值必须比 HIGH 的设定值小; 如 5.3 定义说明。
- 5) 单位为安培 (A)。
- 6) 3330A 专指 DYNAMIC 电阻。

CV: { HIGH|LOW} (适用机型: 3310A, 3310C 系列)

格式: [PRESet:] CV: { HIGH|LOW} {SP} { NR2} { }; |NL}

[PRESet:] CV: { HIGH|LOW}? { }; |NL}

用途: 设定和读取负载电压值

说明: 此命令为设定电子负载的电压值, 下达命令时须注意下列事项:

- 1) 下达的电流值必须含有小数点的数值, 否则命令无效。
- 2) 数值的最小有效位数为小数点后第 5 位。
- 3) 下达的电流数值超过该电子负载的最大规格时, 3302 机框会送出该电子负载规格的满刻度
电流值。
- 4) LOW 的设定电压值必须比 HIGH 的设定电压值小; 如 5.3 定义说明。
- 5) 单位为伏特 (V)。
- 6) 3330A 专指 DYNAMIC 电压。

CV (适用模组: 3330A 系列)

格式: [PRESet:] CV{SP}{ NR2 } { }; |NL}

[PRESet:] CV ? { }; |NL}

用途: 设定和读取负载电压值

说明: 设定电子负载欲载入的静态电压值。

4.6.2 LIMIT设定和读取电子负载判断 NG 的上下限

[LIMit]: CURRent: { HIGH|LOW} (适用机型: 3310C, 3330A, 3250 系列)

格式: [LIMit]: CURRent: { HIGH|LOW}{SP}{NR2};; |NL}

[LIMit]: CURRent: { HIGH|LOW} ? {; |NL}

用途: 设定和读取负载电流的上下限

说明: 设定比较电流的下限值, 当负载 Sink 电流低于此下限值时, NG 指示灯即点亮表示“NO GOOD”。

设定比较电流的上限值, 当负载 Sink 电流高于此下限值时, NG 指示灯即点亮表示“NO GOOD”。

[LIMit]: POWer: { HIGH|LOW} (适用机型: 3250 系列)

格式: [LIMit]: POWer: { HIGH|LOW}{SP}{NR2};; |NL}

[LIMit]: POWer: { HIGH|LOW} ? {; |NL}

用途: 设定和读取负载功率的上下限

说明: 设定比较功率(瓦特)的下限值, 当功率 (瓦特)低于此下限值时, NG 指示灯即点亮表示“NO GOOD”。

设定比较功率(瓦特)的上限值, 当功率 (瓦特)高于此下限值时, NG 指示灯即点亮表示“NO GOOD”。

[LIMit]: VA: { HIGH|LOW} (适用机型: 3250 系列)

格式: [LIMit] VA: { HIGH|LOW}{SP}{NR2};; |NL}

[LIMit] VA: { HIGH|LOW} ? {; |NL}

用途: 设定和读取负载视在功率 (VA) 的上下限

说明: 设定比较功率(伏安)的下限值, 当功率(伏安)低于此下限值时, NG 指示灯即点亮表示“NO GOOD”。

设定比较功率(伏安)的上限值, 当功率(伏安)高于此下限值时, NG 指示灯即点亮表示“NOGOOD”。

[LIMit] VOLtage: { HIGH|LOW} (适用机型: 3310C, 3330A, 3250 系列)

格式: [LIMit] VOLtage: { HIGH|LOW}{SP}{NR2};; |NL}

[LIMit] VOLtage: { HIGH|LOW} ? {; |NL}

用途: 设定和读取负载电压的上下限

说明: 设定比较电压的下限值, 当输入电压低于此下限值时, NG 指示灯即点亮表示“NO GOOD”。

设定比较电压的上限值, 当输入电压高于此下限值时, NG 指示灯即点亮表示“NO GOOD”。

4.6.3 STAGE设定和读取电子负载的工作状态

[STAtE:] LOAD {SP} {ON|OFF} (适用机型: 3310A, 3310C, 3330A, 3250系列)

格式: [STAtE:] LOAD {SP} {ON|OFF} ; |NL}

[STAtE:] LOAD ? ; |NL}

用途: 设定和读取电子负载是否吸收电流

说明: 设定电子负载是否 Sink 电流当设定为 ON 时, 则电子负载开始以待测物 Sink 电流; 当设定为 OFF 时, 则电子负载不会 Sink 电流。

[STAtE:] MODE {SP} {CC|CR|CV|CP} (适用机型: 3310A, 3310C, 3330A, 3250系列)

格式: [STAtE:] MODE {SP} {CC|CR|CV|CP} ; |NL}

[STAtE:] MODE ? ; |NL}

用途: 设定和读取电子负载的操作模式

说明: 电子负载可工作的模式如下表所示

	CC	CR	CV	CP
3310A	3	3	3	
3310C	3	3	3	3
3330A	3	3	3	
3250	3	3		

当读取负载操作模式时, 返回值 0|1|2|3 分别代表 CC|CR|CV|CP 模式

[STAtE:] SHORt {SP} {ON|OFF} (适用机型: 3310A, 3310C, 3330A系列)

格式: [STAtE:] SHORt {SP} {ON|OFF} ; |NL}

[STAtE:] SHORt ? ; |NL}

用途: 设定和读取电子负载是否短路测试

说明: 此命令为设定电子负载作短路测试。当设定为 ON 时, 此时电子负载之 V+,V- 端, 如同短路状态, 其短路阻抗见 3310 系列电子负载使用手册。

[STAtE:] PRESet {SP} {ON|OFF} (适用机型: 3310A, 3310C, 3330A, 3250系列)

格式: [STAtE:] PRESet {SP} {ON|OFF} ; |NL}

[STAtE:] PRESet ? ; |NL}

用途: 设定和读取电子负载是否短路测试

说明: 此命令式控制电流表及 Imonitor 的输出形式。若设为 ON 时, 则电流表及 Imonitor 示之值皆所显示之值皆为预设之值; 若设为 OFF, 则电流表及 Imonitor 所显为实际 SILK 之值。

[STAtE:] SENSE{SP}{ON|OFF} (适用机型: 3310C, 3330A, 3250 系列)

格式: [STAtE:] SENSE{SP}{ON|OFF}{; |NL}

[STAtE:] SENSE ? {; |NL}

用途: 设定和读取电子负载是否短路测试

说明: 此命令为设定电压读取由输入连接器端或是 VSENSE BNC 端, 设定为 ON 时电压值, 由 VSENSE BNC端所取得; 设定为 OFF 时, 电压值是由输入连接器端所取得。

[STAtE:] RANGE {SP}{I II} (适用机型: 3310A, 3320 系列)

格式: [STAtE:] RANGE {SP}{I II}{; |NL}

[STAtE:] RANGE ? {; |NL}

用途: 设定和读取电源或电阻范围

说明: 此命令为设定电流及电阻范围, 3310 系列电子负载共有二档, RANG 1 即 RANG I, RANG 2 即RANGE II, 而其电流及电阻的范围一电子负载模组规格而定, 3310 系列各电子负载模组的电流及电阻范围如下表。

规格 机种	3310		3311		3312		3314		3315	
CC MODE RANGE I/II	0~3A	0~30A	0~6A	0~60A	0~1A	0~10A	0~0.5A	0~5A	0~1.5A	0~15A
CR MODE RANGE II/I	0.1~2Ω	2~8KΩ	0.05~1Ω	1~4KΩ	1.25~25Ω	25~20KΩ	5~100Ω	100~2KΩ	0.2~4Ω	4~16KΩ

[STAtE:] LEVel {SP}{HIGH LOW} (适用机型: 3310A, 3310C, 3250 系列)

格式: [STAtE:] LEVel {SP}{HIGH LOW}; | NL}

[STAtE:] LEVel ? {; | NL}

用途: 设定和读取电子负载 LOW 和 HIGH

说明: 1) LEV LOW 固定电流 (CC) 模式时, 为低准位电流设定值。固定电阻 (CR) 模式时, 为低准位电阻设定值。固定电压 (CV) 模式时, 为低准位电压设定值。

2) LEV 1 固定电流模式时, 为高准位电流设定值。固定电阻模式时, 为高准位电阻设定值。固定电压模式时, 为高准位电压设定值。

[STAtE:] DYNAmic {SP}{ON OFF} (适用机型: 3310A, 3310C, 3330A 系列)

格式: [STAtE:] DYNAmic {SP}{ON OFF}; | NL}

[STAtE:] DYNAmic ? {; | NL}

用途: 设定和读取电子负载为动态或静态负载

说明: 1) DYN ON 设定为动态 (DYNAMIC) 负载。

2) DYN OFF 设定为静态 (STATIC) 负载。

[STAtE:] SYNCRonize {SP}{ON OFF} (适用机型: 3250 系列)

格式: [STAtE:] SYNCRonize {SP}{ON OFF}; | NL}

[STAtE:] SYNCRonize ? {; | NL}

用途: 设定和读取电子负载为外部同步或内部同步

说明: 1) 外部同步信号 (SYNC ON), 使用外部同步信号作为电子负载同步解发信号, 以控制负载电流与电压同步。

2) 内部同步信号 (SYNC OFF), 使用输入连接器的信号, 再经由内部的零交越电路及隔离电路产生同步信号。

[STAtE:] WATT {SP}{ON OFF} (适用机型: 3310C, 3250 系列)

格式: [STAtE:] WATT {SP}{ON OFF}; | NL}

[STAtE:] WATT ? {; | NL}

用途: 设定和读取是否显示功率

说明: 此命令是设定功率电表的显示。本命令必须配合 PRES: OFF 时使用; 设定 ON 时, 上方的显示器由电压表变为瓦特表, 下方的显示器由电流表变为伏安 (VA) 表, 其单位分别为, “W”, “VA”。

设定 OFF 时, 上方的瓦特 (W) 变回电压表, 下方的伏安 (VA) 表变回电流表, 其单位分别为 “Vrms”, “Arms”。

[STAtE:] CLRrerr (适用机型: 3310C, 3330A, 3250 系列)

格式: [STAtE:] CLRrerr {; | NL}

用途: 清除当前模组在工作过程中产生的错误标志

说明: 此命令为清除 PROT 及 ERR 暂存器内容, 执行后 PROT 及 ERR 暂存器内容全部为 “0”。

STAtE:]ERRor (适用机型: 3310C, 3330A, 3250 系列)

格式: [STAtE:]ERRor {; | NL}

用途: 查询当前模组是否有的错误标志

说明: 1) ERR ? 读回错误暂存器 (ERR) 的状态, 下表说明错误状态的位元对应码。

2) ERR 状态暂存器的清除, 可以使用 CLEAR 命令将 ERR 状态暂存器清除为“0”。

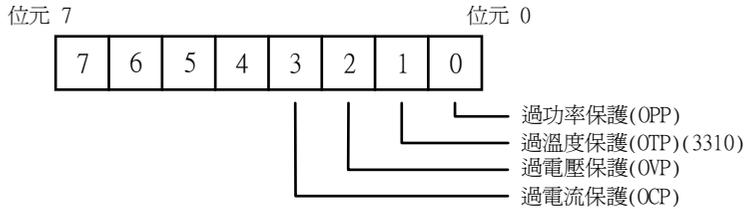


表4.3 3250 系列 ERR 状态暂存器

位元 0 : 发生限制动作

此位元码设定为“1”时, 表示发生了设定值超过规格的最大值, 此时 3250 负载会自动将设定值限制在规格的最大值内。

位元 1 : 发生换档动作

此位元码设定为“1”时, 表示发生了换档的动作, 由于 3250 系列拥有自动换档的能力, 所以此位元码仅表示曾经有换档的动作而非错误发生。

位元 2 : 不正确的命令

此位元设定为“1”时, 表示系统接收到一个不正确的命令。

位元 3 : 不正确的操作

此位元设定为“1”时, 表示系统接收到一个不正确的操作, 例如: 在固定电阻 (CR) 模式下企图设定 C.F. 值 (WAVE 命令)。

[STAtE:]DUAL {DVM DVM OFF (适用机型: 3330A 系列)}

]格式: [STAtE:]DUAL {DVM DVM OFF} {; | NL}

用途: 设定当前模组在显示状态

说明: 此命令为设定 3330A 系列模组在显示状态:

DVM: 雙电压显示模式

DAM: 雙电压显示模式

OFF: NORMAL显示模式

[STAtE:]NG? (适用机型: 3310C, 3330A, 3250 系列)

]格式: [STAtE:]NG? {; | NL}

用途: 查询当前模组是否有的 NG 标志

说明: NG? 读回NG 的状态指示灯, “0”表示 NG (NO GOOD) 指示灯熄灭, “1”表示 NG 指示灯点亮。

[STAtE:]PROTect? (适用机型: 3310A, 3310C, 3330A, 3250 系列)

]格式: [STAtE:]PROTect? {; | NL}

用途: 查询当前模组是否有的保护标志

说明: 1) PROT ? 读回负载目前的保护状态, “1”表是发生 OPP, “4”表示发生 OVP, “8”表示发生 OCP, 下表说明保护状态位元对应码。

2) PROT 状态暂存器的清除, 可以使用 CLER 命令将 PROT 状态暂存器清除为 “0”。

4.6.4 SYSTEM设定和读取机框和电子负载的状态

[SYStem:] RECall{ SP }m{ ,n } (适用机型: 3310A, 3310C, 3330A, 3250 系列)

格式: [SYStem:] RECall{ SP }m{ ,n }{ ; | NL }

用途: 呼叫记忆装定中的负载状态

说明: 此命令为呼叫记忆装定中的负载状态资料, m=1~5, n=1~30

如果当前模组为 3310A, 3250 系列时, 项目号 (n) 无效, n 可以省略。

如果当前模组为其他系列时, 省略符号n, 项目默议为 3300C 所显示的项目。

实例: RECALL 2, 15 呼叫记忆装定中的第2组第15项负载装态资料。

REC 3 : 呼叫记忆装定中的第3组负载状态资料, 如果当前模组为3320, 3310C, 3330A 系列时, 项目默议为 3300C 所显示的项目。

[SYStem:] STORe{SP}m{n} (适用机型: 3310A, 3310C, 3330A, 3250 系列)

格式: [SYStem:] STORe{SP}m{n}{ ; | NL }

用途: 存储负载状态到记忆装定中

说明: 此命令为存储负载状态到记忆装定中, m=1~5, n=1~30; 如果当前模组为 3310A, 3250 系列时, 项目号 (n) 无效, n 可以省略。

如果当前模组为其他系列时, 省略项目号 n, 项目默议为 3300C 所显示的项目。

实例: SYSTEM: STORE 2, 15 存储负载状态到记忆装定第 2 组第 15 项中

STOR 3: 存储负载状态到记忆装定第 3 组, 如果当前模组为3320, 3310C, 3330A 系列时, 项目默议为 3300C 所显示的项目。

[SYStem:] NAME ? (适用机型: 3310A, 3310C, 3330A, 3250 系列)

格式: [SYStem:] NAME ? { ; | NL }

用途: 读取当前电子负载机框编号

说明: 此命令读回当前电子负载机框编号, 如果当前没有模组, 那麽将读到字符串“NONE”; 如果当前安装了模组, 那麽将会读到以下型号:

型号	型号	型号	型号
3250	3310A	3310C	3330A
3251	3311A	3311C	3331A
3252	3312A	3312C	3332A
3253	3314A	3314C	3333A
	3315A	3315C	3334A
			3335A

4.6.5 MEASURE 测量电子负载的当前电流电压的实际值

MEASure: CURRent ? (适用机型: 3310A, 3310C, 3330A, 3250 系列)

格式: MEASure: CURRent{? | NL}

用途: 读取当前电子负载的电流

说明: 读回 4 位半数位电流表的读值, 单位为安培 (A)

MEASure: VOLtage ? (适用机型: 3310A, 3310C, 3330A, 3250 系列)

格式: MEASure: VOLtage{? | NL}

用途: 读取当前电子负载的电压

说明: 读回 4 位半数位电压表的读值, 单位为伏特 (V)

4.6.6 OFFSET GAIN 校正电子负载

OFFSet : {CC CR}: I: {AC DC} (适用机型: 3250 系列)

格式: OFFSet : {CC CR}: I: {AC DC}{SP}{NR3}{; | NL}

用途: 修改当前电子负载的 RANGE I AC DC 预制电流 | 预定电阻的零点

说明: 3250 校正项目 1, NR3= -0C18 ~ 03E8

GAIN: {CC CR}: I: {AC DC} (适用机型: 3250 系列)

格式: GAIN: {CC CR}: I: {AC DC}{SP}{NR3}{; | NL}

用途: 修改当前电子负载的 RANGE I AC DC 预制电流 | 预定电阻的增益

说明: 3250 校正项目 2, NR3= -0898 ~ 0708

OFFSet: {CC CR}: II: {AC DC} (适用机型: 3250 系列)

格式: OFFSet: {CC CR}: II: {AC DC}{SP}{NR3}{; | NL}

用途: 修改当前电子负载的 RANGE II AC DC 预制电流 | 预定电阻的零点

说明: 3250 校正项目 3, NR3= 0C18 ~ 03E8

GAIN: {CC CR}: II: {AC DC} (适用机型: 3250 系列)

格式: GAIN: {CC CR}: II: {AC DC}{SP}{NR3}{; | NL}

用途: 修改当前电子负载的 RANGE II AC DC 预制电流 | 预定电阻的增益

说明: 3250 校正项目 4, NR3= 0E10 ~ 0FFF

OFFSet: MEASure: VOLtage: {AC DC} (适用机型: 3250 系列)

格式: OFFSet: MEASure: VOLtage: {AC DC}{SP}{NR3}{; | NL}

用途: 修改当前电子负载的 AC DC 电压表的零点

说明: 3250 校正项目 5, NR3= 0830 ~ 07D0

GAIN: MEASure: VOLtage: {AC DC} (适用机型: 3250 系列)

格式: GAIN: MEASure: VOLtage: {AC DC}{SP}{NR3}{; | NL}

用途: 修改当前电子负载的 AC DC 电压表的增益

说明: 3250 校正项目 6, NR3= 0650 ~ 05F0

OFFSet: MEASure: CURRent: {AC DC} (适用机型: 3250 系列)

格式: OFFSet: MEASure: CURRent: {AC DC}{SP}{NR3}{; | NL}

用途: 修改当前电子负载的 AC DC 电流表的零点

说明: 3250 校正项目 7, NR3= -0830 ~ 07D0

OFFSet: MEASure: CURRent: {AC DC} (适用机型: 3250 系列)

格式: OFFSet: MEASure: CURRent: {AC DC}{SP}{NR3}{; | NL}

用途: 修改当前电子负载的 AC DC 电压表的增益

说明: 3250 校正项目 8, NR3= 0650 ~ 05F0

OFFSet: MEASure: POWer: {AC DC} (适用机型: 3250 系列)

格式: OFFSet: MEASure: POWer: {AC DC}{SP}{NR3}{; | NL}

用途: 修改当前电子负载的AC DC WATT表的零点

说明: 3250 校正项目 9, NR3=- 0830 ~ 05F0

GAIN: MEASure: POWer: {AC DC} (适用机型: 3250 系列)

格式: GAIN: MEASure: POWer: {AC DC}{SP}{NR3}{; | NL}

用途: 修改当前电子负载的 AC DC WATT表的增益

说明: 3250 校正项目 10, NR3= 0650 ~ 05F0

{OFFSet GAIN}: CC: {I II}: {HIGH LOW} (适用机型: 3310C, 3330A 系列)

格式: {OFFSet GAIN}: CC: {I II}: {HIGH LOW}{SP}{NR3}{; | NL}

用途: 修改当前电子负载的 RANGE I II HIGH LOW 预定功率的零点或增益

说明:

{OFFSet GAIN}: CP: I: {HIGH LOW} (适用机型: 3310C 系列)

格式: {OFFSet GAIN}: CP: I: {HIGH LOW}{SP}{NR3}{; | NL}

用途: 修改当前电子负载的RANGE I HIGH LOW HIGH LOW预定电流的零点或增益

说明:

{OFFSet GAIN}: CC: {I II}: {HIGH LOW} (适用机型: 3310C 系列)

格式: {OFFSet GAIN}: CC: {I II}: {HIGH LOW}{SP}{NR3}{; | NL}

用途: 修改当前电子负载的 RANGE I II HIGH LOW 预定电阻的零点或增益
说明:

{OFFSet GAIN}: CC: {SP}{NR3} (适用机型: 3330A 系列)

格式: {OFFSet GAIN}: CC: {SP}{NR3}{; | NL}

用途: 修改当前电子负载的 RANGE I 预定电压的零点或增益

说明:

{OFFSet GAIN}: CV: I{SP}{NR3} (适用机型: 3330A 系列)

格式: {OFFSet GAIN}: CV: I{SP}{NR3}{; | NL}

用途: 修改当前电子负载的 RANGE I 预定电压的零点或增益

说明:

{OFFSet GAIN}: CV: {I II}: {HIGH LOW}{SP}{NR3}

(适用机型: 3310C系列)

格式: {OFFSet GAIN}: CV: {I II}: {HIGH LOW}{SP}{NR3}{; | NL}

用途: 修改当前电子负载的 RANGE I II HIGH LOW 预定电压的零点或增益

说明:

{OFFSet GAIN}: MEASure: VOLtagte: {I II}: {SP}{NR3}

(适用机型: 3310C, 3330A 系列)

格式: {OFFSet GAIN}: MEASure: VOLtagte: {I II}: {SP}{NR3}{; | NL}

用途: 修改当前电子负载的 RANGE I II 预定电流的零点或增益

说明:

{OFFSet GAIN}: MEASure: VOLtagte: SENSE: {I II}: {SP}{NR3}

(适用模组: 3310C 系列)

格式: {OFFSet GAIN}: MEASure: VOLtagte: SENSE: {I II}: {SP}{NR3}{; | NL}

NL}

用途: 修改当前电子负载的 SENSE 的电压表 RANGE I II 预定电流的零点或增益

说明:

附录1 3310A 系列 GPIB 程式范例

C 语言程式范例

```
/* Link this program with appropriate *cib*.obj. */
```

```
/* This application program is written in TURBO C 2.0 for the IBM PC-AT compatible. The National Instruments Cooperation (NIC) Model PC-2A board provides the interface between the PC-AT and a PRODIGIT MPAL ELECTRONIC LOAD. The appropriate *cib*.obj file is required in each program to properly link the NIC board to C LANGUAGE. and include the <decl.h.> HEADER FILE to C LANGUAGE. */
```

```
#include <stdio.h>
```

```
#include <dos.h>
```

```
#include <math.h>
```

```
#include "decl.h" /* NI GPIB CARD HEADER FILE */
```

```
main()
```

```
{
```

```
    char ouster[20],rdbuf[15],spec[10];
```

```
    int i,ch,load;
```

```
/* Assign unique identifier to the device "dev5" and store in variable load. check for error. ibfind error = negative value returned. */
```

```
    if((load = ibfind("dev5")) < 0) /* Device variable name is load */
```

```
    { /* GPIB address is 5 */
```

```
        printf("\r*** INTERFACE ERROR! ***\a\n");
```

```
        printf("\r\nError routine to notify that ibfind failed.\n");
```

```
        printf("\r\nCheck software configuration.\n");
```

```
        exit(1);
```

```
    }
```

```
/* Clear the device */
```

```
    if((ibclr(load)) & ERR);
```

```
    {
```

```
        printf("INTERFACE ERROR! \a");
```

```
        exit(1);
```

```
    }
```

```
    clrscr();
```

```
/* Clear load error register */
```

```
    for(i=1,ch=0;i<=4;i++,ch++)
```

```
{
  outstr=chan[ch];
  ibwrt(load,outstr,6);
  ibwrt(load,"CLER",4);
}
ibwrt( load,"NAME?",5);          /* Get the 3310A series module load specification */
strset(rdbuf,'\0');            /* Clear rdbuf string buffer */
strset(spec,'\0');             /* Clear spec string buffer */
ibrd(load,spec,20);
if (spec[3] == '9')
  printf("\n 3300C series specification error !");
/* Set the channel 1, preset off, current sink 1.0 amps and load on commands to the load. */
ibwrt( load,"chan 1;pres off;curr:low 0.0;curr high 1.0;load on ",51);
ibwrt( load,"meas:curr ?",11);
/* Get the load actually sink current from the load */
ibrd( load,rdbuf,20);
/* go to local. */
ibloc(load);
```

BASICA 语言程式范例

LOAD DECL.BAS using BASICA MERGE command.

```
100 REM You must merge this code with DECL.BAS
105 REM
110 REM Assign a unique identifier to the device "dev5" and store it in variable load%.
125 REM
130     udname$ = "dev5"
140     CALL ibfind (udname$,load%)
145 REM
150 REM Check for error on ibfind call
155 REM
160     IF load% < 0 THEN GOTO 2000
165 REM
170 REM Clear the device
175 REM
180     CALL ibclr (load%)
185 REM
190 REM Get the 3310A series module load specification
195 REM
200     wrt$ = "NAME?" : CALL ibwrt(load%,wrt$)
210     rd$ = space$(20) : CALL ibrd(load%,rd$)
215 REM
220 REM Set the channel 1, preset off, current sink 1.0 amps and load on commands to the load.
225 REM
230     wrt$ = "chan 1;pres off;curr:low 0.0;curr high 1.0;load on"
240     CALL ibwrt(load%,wrt$)
245 REM
250 REM Get the load actually sink current from the load
255 REM
260     wrt$ = "meas:curr?" : CALL ibwrt(load%,wrt$)
270     rd$ = space$(20) : CALL ibrd(load%,rd$)
275 REM
280 REM Go to local
285 REM
290 CALL ibloc(load%)
```

```
2000 REM Error routine to notify that ibfind failed.  
2010 REM Check software configuration.  
2020 PRINT "ibfind error !" : STOP
```

附录2 3310A 系列 RS-232 程式范例

C Language Interface for DOS Handlers " pd_rs232.c "

```

#include <dos.h>
#include <stdio.h>
#include <conio.h>
#include <string.h>

#define COMPTR          44          /*      command      array      pointer
*/

#define QUELEN          1024        /* size of serial input */
#define SUCCESS         0          /* return value variable for success */
#define OPER_ERR        -1         /*      operate      error
*/

#define TIME_OUT        1          /*      time_out
*/

int input_index = 0;              /* index of serial input buffer */
int rd_result = 0;               /* return value variable of pd_rd() */
int timeout = 0;                 /* timeout flag */
char queue[QUELEN];             /*      serial      input      buffer
*/
int ACE_DATA_REG;               /*      ACIA      data      register
*/
int ACE_INT_ENB_REG;            /* ACIA interrupt enable register */
int ACE_INT_IDENT_REG;         /* ACIA interrupt identification register*/
int ACE_LINE_CTL_REG;          /*      ACIA      line      control      register
*/
int ACE_MODEM_CTL_REG;         /*      ACIA      modem control register
*/
int ACE_LINE_STAT_REG;         /* ACIA line status register */
int ACE_MODEM_STAT_REG;        /*      ACIA      modem      status      register
*/
int COM_INT_NUM;               /* ACIA communication port interrupt
number*/
int IRQ_MASK;                  /*      IRQ      mask      for      PC      IRQ      flag
*/

void pd_loc(void);              /* function of disable interrupt routine */

```

```

void pd_rem(void); /* function of enable interrupt routine */
int pd_init(int); /* function of initial communication port*/
int pd_wrt(char *,int); /* function of write to device */
int pd_rd(char *,int); /* function of read from device */
/*
int read_buf(void);
int pd_meas(char *,char*, int, int);

#define PIC_CTL_REG 0x20 /* 8259A PIC control register */
#define PIC_INT_MASK_REG 0x21 /* 8259A PIC interrupt mask register */
#define NON_SPEC_EOI 0x20 /* non-specific end of interrupt */
#define CTS 0x10 /* clear to send */
#define DSR 0x20 /* data set ready */
#define RI 0x40 /* ring indicator */
#define DCD 0x80 /* data carrier detect */
#define DCTS 1 /* delta clear to send */
#define DDSR 2 /* delta data set ready */
#define TERI 4 /* trailing edge ring detect */
#define DDCD 8 /* delta data carrier detect */
#define OE 2 /* overrun error */
#define PE 4 /* parity error */
#define FE 8 /* frame error */
#define BI 0x10 /* break interrupt */
#define THRE 0x20 /* transmit holding reg. empty */

/* 3300C series elec. load command sets */
int pd_wrt(char *wrtbuf,int count)
{
    static char *combuf[COMPTR] = {"CHAN", "CURR:HIGH", "CURR:LOW",
    "RES:HIGH", "RES:LOW", "VOLT:HIGH",
    "VOLT:LOW", "PERD:HIGH", "PERD:LOW",
    "FALL", "RISE", "GLOB:LOAD",
    "GLOB:LEV", "GLOB:PRES", "GLOB:SHOR",
    "GLOB:DYN", "GLOB:RANG", "GLOB:MODE",
    "CLER", "CHAN?", "MEAS:VOLT?",
    "MEAS:CURR?","CURR:HIGH?","CURR:LOW?",
    "ERR?", "RES:HIGH?", "RES:LOW?",
    "VOLT:HIGH?","VOLT:LOW?", "NAME?",
    "PERD:HIGH?","PERD:LOW?", "FALL?",

```

```

"RISE?",      "LOAD?",      "LEV?",
"PRES?",      "SHOR?",      "DYN?",
"RANG?",      "MODE?",      "PROT?",
"REMOTE",     "LOCAL" };

int cnt,result,t;
int comerr,err;
char ch;
char tempbuf[QUELEN];
char intbuf[QUELEN];
/*   for (cnt = 0;cnt < 1024;cnt++)
{
    intbuf[cnt] = "";
    tempbuf[cnt]="";
}   */
strset(intbuf,"");
strset(tempbuf,"");
for (cnt = 0;cnt < count+1;cnt++,wrtbuf++)
{
    intbuf[cnt] = *wrtbuf;
}
cnt = cnt--;
intbuf[cnt] = '\r';
cnt = cnt++;
count = count++;
intbuf[cnt] = '\n';
t = strcspn(intbuf, " ");
if (t > count)
t = count-1;
for (cnt = 0;cnt < t;cnt++)
{
    tempbuf[cnt] = intbuf[cnt];
}
strupr(tempbuf);
strupr(intbuf);
cnt = 0;
do
{
    comerr = strncmp(tempbuf,combuf[cnt],t);

```

```

    cnt = cnt++;
}while(( cnt != COMPTR) && (comerr != 0));
if ( comerr == 0)
{
    result = SUCCESS;
for (cnt = 0;cnt < count+1;cnt++)
{   while(inportb(ACE_LINE_STAT_REG) & THRE == 0);
    ch = intbuf[cnt];
    delay(20);
    outportb(ACE_DATA_REG, ch);
}
timeout = 0;
for (cnt = 0; cnt <= count;cnt++)
{
    ch = intbuf[cnt];
    if(ch == '?')
    {
        do
        {
            read_buf();
        }while((rd_result == 0)&&(timeout == 0));
        rd_result = 0;
    }
}
for (cnt = 0;cnt<= count;cnt++)
{
    intbuf[cnt] = "";
    tempbuf[cnt] = "";
}
return(result);
}
return(OPER_ERR);
}
int pd_rd(char *buf,int count)
{
    char ch;
    int cnt = 0;
    do

```

```

    {
        ch = queue[cnt];
        *buf = ch;
        cnt = cnt++;
        buf = buf++;
    }while((ch != '\n') && (cnt != count));
    *buf = '\0';
    queue[0] = '\0';
    if (timeout == 1)
    {
        return(TIME_OUT);
    }
    return(SUCCESS);
}
int read_buf()
{
    char ch,ch1;
    int temp_index;
    unsigned long ticks;
    float sec1,sec2;
    rd_result = 0;
    ticks = biostime(0,0);
    sec1 = ticks/18.2;
    do
    {
        ticks = biostime(0,0);
        sec2 = ticks/18.2;
        if((sec2 - sec1) >= 20.0)                /* delay about 1 Sec */
        {
            timeout = 1;
        }
        ch = inportb(ACE_INT_IDENT_REG); ch &= 0x06;
        switch(ch)
        {
            case 6:
                inportb(ACE_DATA_REG);                /* read the data register to empty
it */
                break;
            case 0:

```

```

        break;
    case 2:
        break;
    case 4:
        /* read character from data
register */
        ch1 = inportb(ACE_DATA_REG);
        temp_index = input_index + 1; /* increment index of input buffer*/
        if (ch1 != '\n') /* check terminate bit */
        {
            queue[input_index] = ch1; /* store character to input buffer*/
            input_index = temp_index;
        }
        else
        {
            queue[input_index] = ch1;
            input_index = temp_index;
            input_index = 0; /* if terminate bit was detected */
            temp_index = 0;
            rd_result = 1; /* clear index and set return value */
        }
        break;
    } /* finally send the non-specific
*/
} while ((rd_result == 0) && (timeout == 0));
}

int pd_init(int pd_com)
{
    if ((pd_com != 1) && (pd_com != 2))
        return(OPER_ERR);
    if (pd_com == 2) /* initial communication port 2
*/
    {
        ACE_DATA_REG = 0x2f8;
        ACE_INT_ENB_REG = 0x2f9;
        ACE_INT_IDENT_REG = 0x2fa;
        ACE_LINE_CTL_REG = 0x2fb;
        ACE_MODEM_CTL_REG = 0x2fc;
        ACE_LINE_STAT_REG = 0x2fd;
    }
}

```

```

    ACE_MODEM_STAT_REG = 0x2fe;
    COM_INT_NUM        = 11;
    IRQ_MASK           = 0xf7;          /* IRQ mask for IRQ3
(11110111) */
    }
    else                /* initial communication port 1
*/
    {
        ACE_DATA_REG      = 0x3f8;
        ACE_INT_ENB_REG   = 0x3f9;
        ACE_INT_IDENT_REG = 0x3fa;
        ACE_LINE_CTL_REG  = 0x3fb;
        ACE_MODEM_CTL_REG = 0x3fc;
        ACE_LINE_STAT_REG = 0x3fd;
        ACE_MODEM_STAT_REG = 0x3fe;
        COM_INT_NUM        = 12;
        IRQ_MASK           = 0xf6;          /* IRQ mask for IRQ4
(11101111) */
    }
    bioscom(0, 0xc3, pd_com-1);          /* boud rate : 9600, 1 start bit
*/
    pd_rem();                             /* no parity, 1 stop bit.
*/
    pd_wrt("remote",6);                   /* data bit : 8 bits */
    return(SUCCESS);
}

```

```
int pd_meas(char *wrtbuf, char *rdmbuf,int wrtmcnt,int rdmcnt)
```

```

{
    static char *wrtbuf[10] = { "CURR 1", "CURR 2", "CURR 3", "CURR 4",
                                "VOLT 1", "VOLT 2", "VOLT 3", "VOLT 4", }
    char tmpbuf[20] = {"chan "};
    int cnt,result,t,cmp,measerr;
    char ch;
    char rdtmpbuf[20],chanbuf[20];
    char measbuf[6];
    for (cnt = 0; cnt < wrtmcnt; cnt++, wrtbuf++)
    {
        measbuf[cnt] = *wrtbuf;
    }
}

```

```
}
strupr(measbuf);
cnt = 0;
do
{
    measerr = strncmp(measbuf,wrtbuf[cnt],6);
    cnt = cnt++;
}while((cnt != 10) && (measerr != 0));
if(measerr == 0)
{
    result = SUCCESS;
    if (strncmp(measbuf,"CURR", 4) == 0)
    {
        ch = measbuf[5];
        switch(ch)
        {
```

```

    case '1':
        pd_wrt("chan 1",6);
        pd_wrt("meas:curr?",10);
        pd_rd(rdmbuf,rdmcnt);
        strcat(tmpbuf,chanbuf,1);
        pd_wrt(tmpbuf,6);
        break;
    case '2':
        pd_wrt("chan 2",6);
        pd_wrt("meas:curr?",10);
        pd_rd(rdmbuf,rdmcnt);
        strcat(tmpbuf,chanbuf,1);
        pd_wrt(tmpbuf,6);
        break;
    case '3':
        pd_wrt("chan 3",6);
        pd_wrt("meas:curr?",10);
        pd_rd(rdmbuf,rdmcnt);
        strcat(tmpbuf,chanbuf,1);
        pd_wrt(tmpbuf,6);
        break;
    case '4':
        pd_wrt("chan 4",6);
        pd_wrt("meas:curr?",10);
        pd_rd(rdmbuf,rdmcnt);
        strcat(tmpbuf,chanbuf,1);
        pd_wrt(tmpbuf,6);
        break;
    }
}
else if(strncmp(measbuf,"VOLT",4) == 0)
{
    ch = measbuf[5];
    switch(ch)
    {
    case '1':
        pd_wrt("chan 1",6);
        pd_wrt("meas:volt?",10);

```

```
        pd_rd(rdmbuf,rdmcnt);
        strncat(tmpbuf,chanbuf,1);
        pd_wrt(tmpbuf,6);
        break;
case '2':
        pd_wrt("chan 2",6);
        pd_wrt("meas:volt?",10);
        pd_rd(rdmbuf,rdmcnt);
        strncat(tmpbuf,chanbuf,1);
```

```

        pd_wrt(tmpbuf,6);
        break;
    case '3':
        pd_wrt("chan 3",6);
        pd_wrt("meas:volt?",10);
        pd_rd(rdmbuf,rdmcnt);
        strcat(tmpbuf,chanbuf,1);
        pd_wrt(tmpbuf,6);
        break;
    case '4':
        pd_wrt("chan 4",6);
        pd_wrt("meas:volt?",10);
        pd_rd(rdmbuf,rdmcnt);
        strcat(tmpbuf,chanbuf,1);
        pd_wrt(tmpbuf,6);
        break;
    }
}
else
{
    return(OPER_ERR);
}
return (result);
}

void pd_rem(void)
{
    char ch;
    outportb(ACE_INT_ENB_REG, 0xd);          /* enable ACIA interrupt register
*/
    inportb(ACE_DATA_REG);                  /* empty receive data register
*/
    inportb(ACE_LINE_STAT_REG);            /* clear line status register
*/
    outportb(ACE_MODEM_CTL_REG, 0xb);      /* set RTS,DTR to enable modem
and */
                                           /* turn on OUT2 to enable the
8250's */
}

```

```

/* IRQ interrupt to system
*/
}
void pd_loc(void)
{
    char ch;
    outportb(ACE_INT_ENB_REG, 0); /* disable all 8250 interrupt */
    outportb(ACE_MODEM_CTL_REG, 0); /* clear RTS,DTR to disable
modem and */
/* turn off OUT2 to disable the
8250's*/
/* IRQ interrupt to system
*/
}/* Program terminated. */

```

C Example Program

```

/* Link this program with pd_rs232.obj */

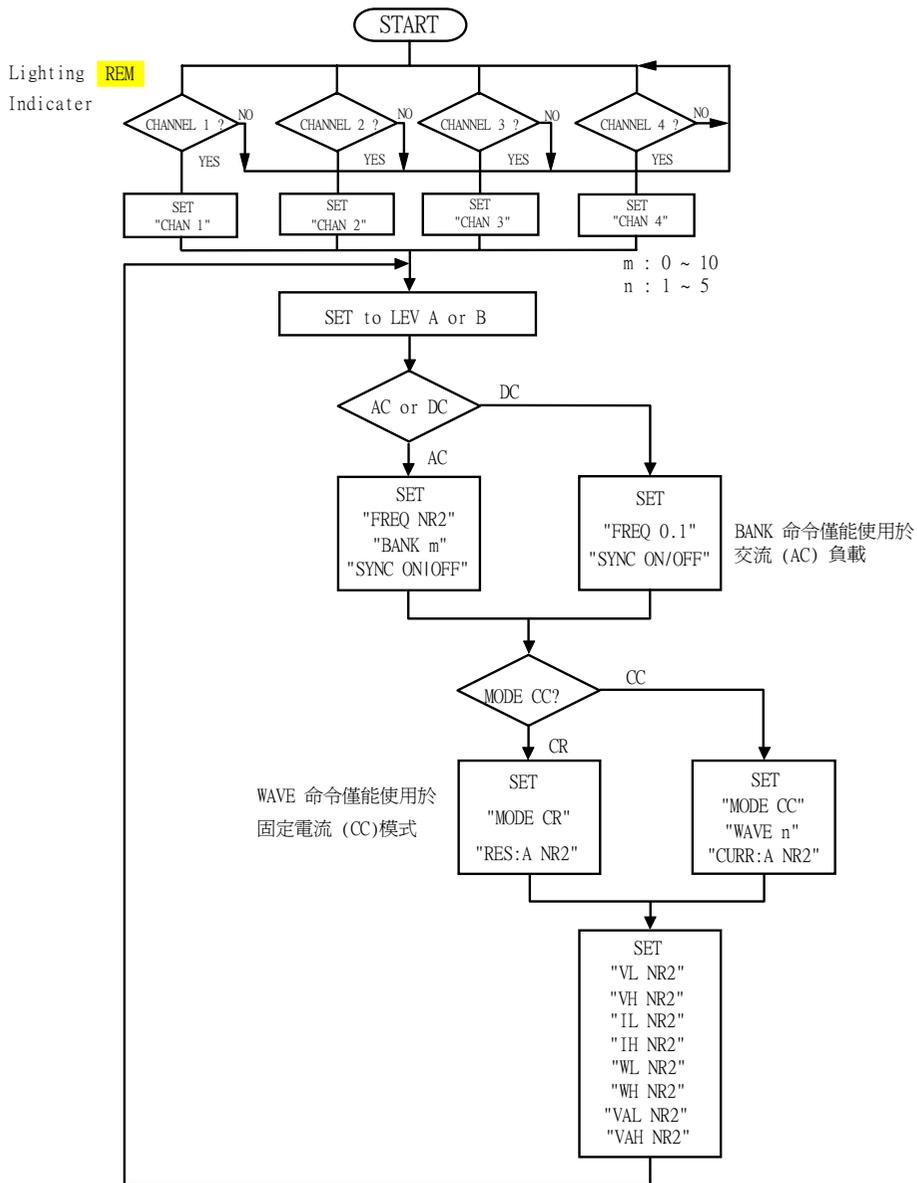
#include <dos.h>
#include <stdio.h>
#include <conio.h>
#include <string.h>

main()
{
    int com;
    int io_err= 0;
    char rdbuf[1024];
    clrscr();
    printf("Input COM1/COM2 port is : ");
    scanf("%d", &com);
    if((io_err = pd_init(com)) == 0) /* Initial RS-232 interface */
    {
        setstr(rdbuf,"");
        pd_wrt("chan 1",6);
        delay(200);
        pd_wrt("name?",5);
        if ((io_err = pd_rd(a,10)) == 0)

```

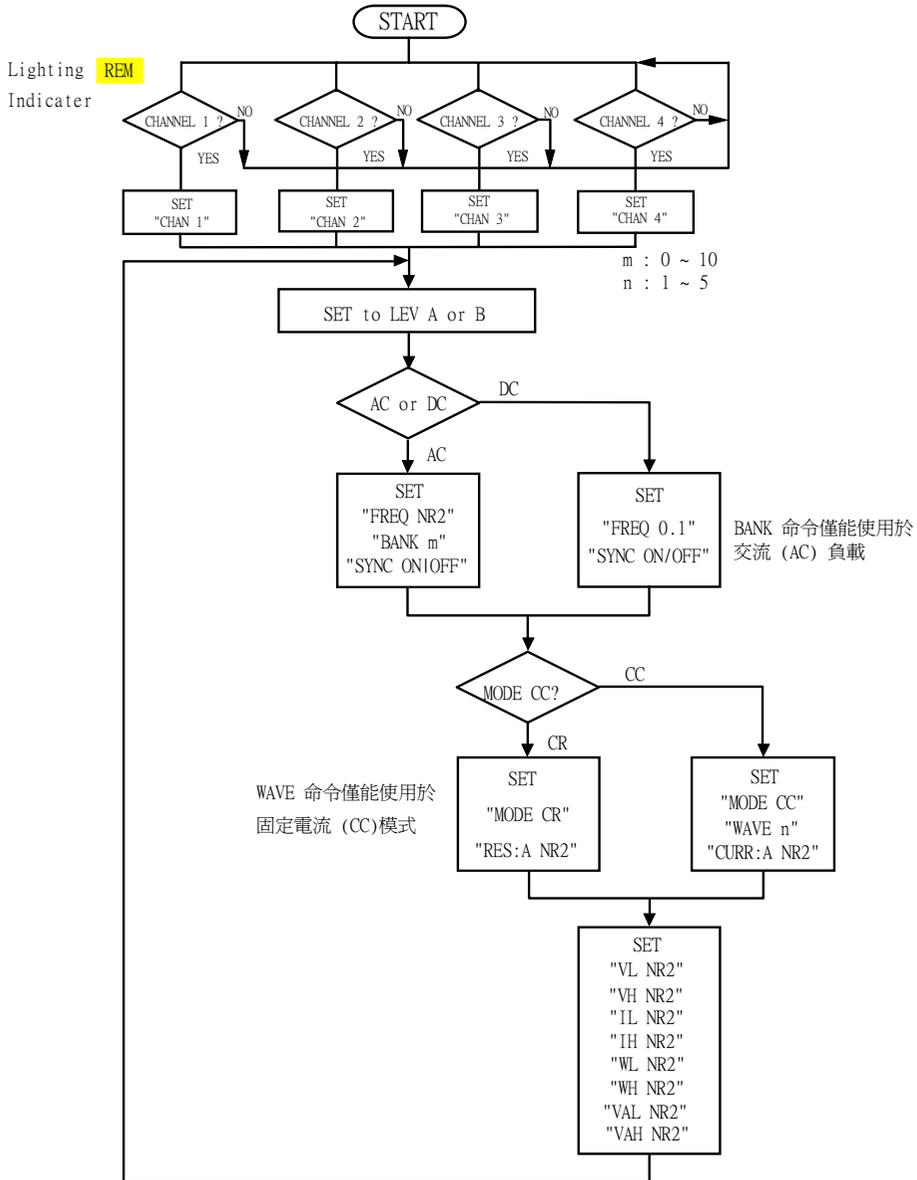
```
{
    do
    {
        /* Set the channel 1, preset off, current sink 1.0 amps and load on commands to the load. */
        pd_wrt("chan 1",6);
        delay(200);
        pd_wrt("pres off",8);
        delay(200);
        pd_wrt("curr:low 0.0",12);
        delay(200);
        pd_wrt("curr high 1.0",13);
        delay(200);
        pd_wrt("load on",7);
        delay(200);
        pd_wrt("meas:curr ?",11);
        /* Get the load actually sink current from the load */
        delay(200);
        pd_rd(rdbuf,20);
        io_err = 1;
    }while (io_err == 0);
}
else
{
    printf("\a");
    printf("chan 1 I/O reading error !\n");
    exit(1);
}
pd_loc(); /* Go to local */
}
```

附录3 3250 系列 GPIB/RS-232C 操作流程圖



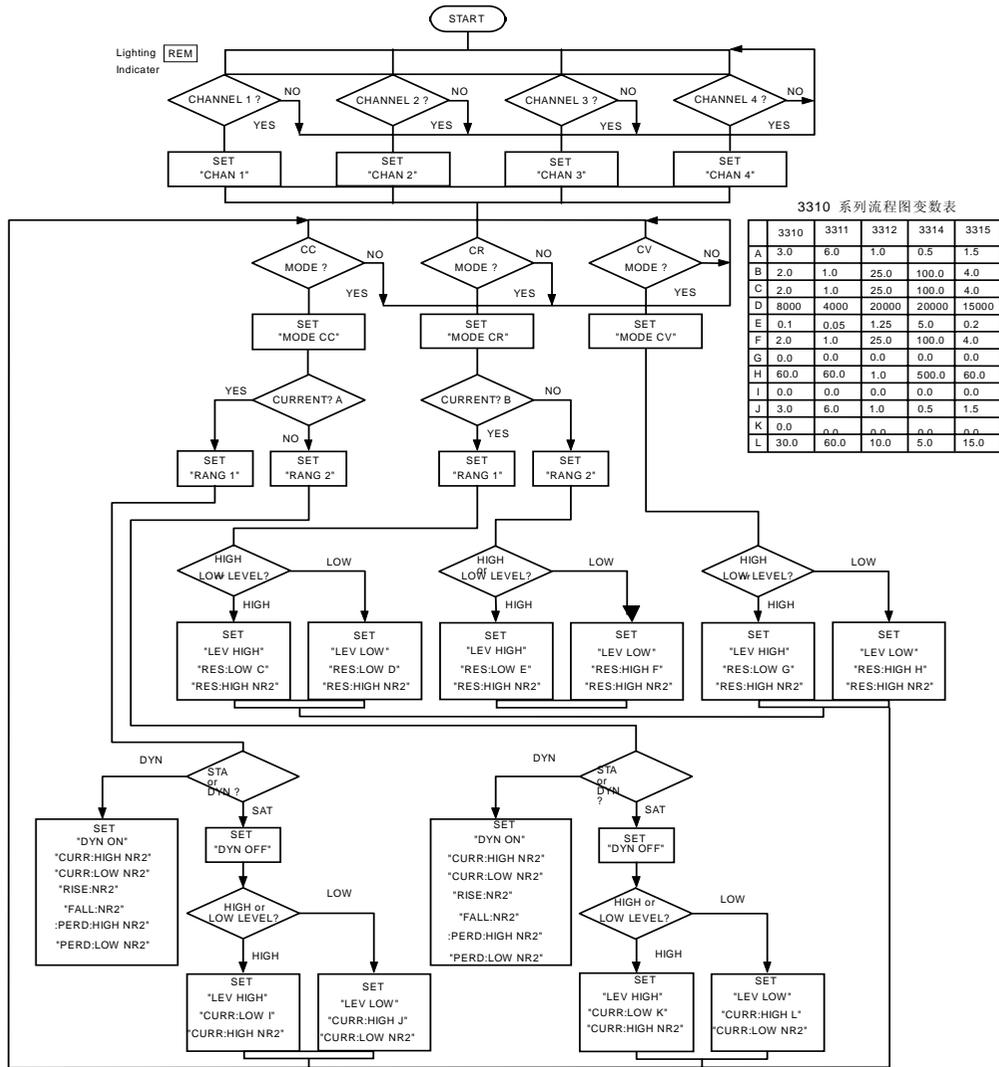
3250 系列 GPIB/RS-232C 操作流程圖

附录3 3250 系列 GPIB/RS-232C 操作流程



3250 系列 GPIB/RS-232C 操作流程

附录4 3310A 系列 GPIB/RS-232C 操作流程



3310A 系列 GPIB 控制流程图



北京海洋兴业科技股份有限公司 (证券代码: 839145)

北京市西三旗东黄平路19号龙旗广场4号楼(E座)906室

电话: 010-62176775 62178811 62176785

企业QQ: 800057747 维修QQ: 508005118

企业官网: www.hyxyyq.com

邮编: 100096

传真: 010-62176619

邮箱: market@oitek.com.cn

购线网: www.gooxian.net



扫描二维码关注我们
查找微信公众号: 海洋仪器