

# CF-TP 系列 交流電源供應器

## 操作手冊



## 目錄：

壹 · 電氣規格.....	1-1
貳 · 配線說明.....	2-1
參 · 面板說明.....	3-1
肆 · 背板說明.....	4-1
伍 · 操作程序.....	5-1
陸 · GPIB、RS232 界面指令說明.....	6-1
柒 · 錯誤碼對照表.....	7-1
捌 · RS232 鮑率與 GPIB 位址設定法.....	8-1
玖 · 程式範例.....	9-1

## 附錄：

壹 · 輸出特性曲線圖.....	A1-1
貳 · 推動板故障檢修.....	A2-1
參 · 導線線徑與電流規格表.....	A3-1

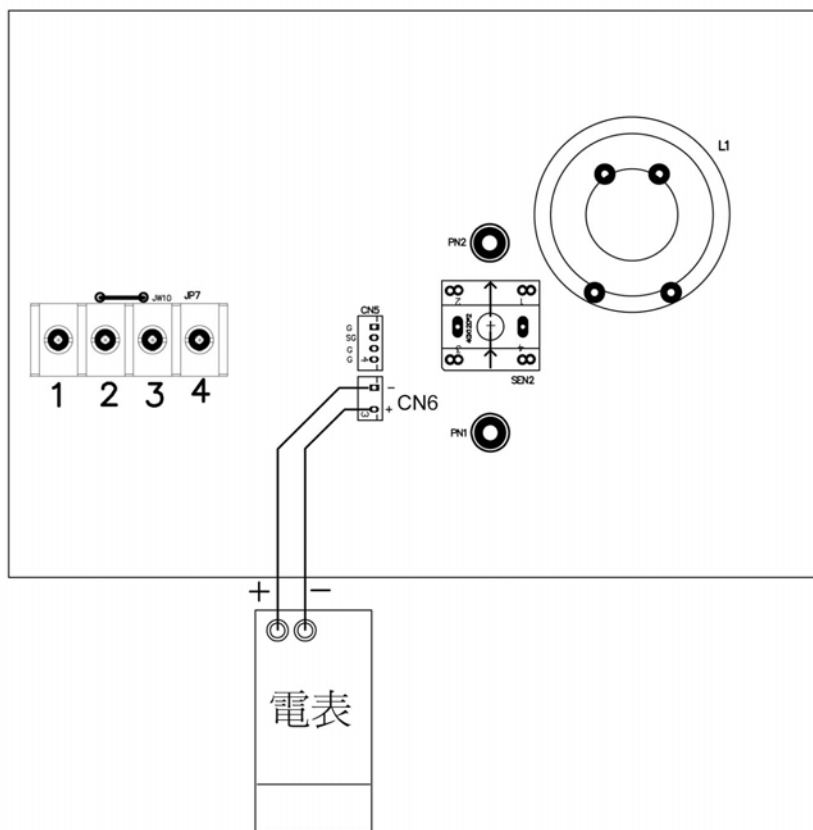
# 壹.電氣規格

1. 輸入電壓 : AC 110V/AC 220V  $\pm 10\%$  1 $\emptyset$ 2W 50/60Hz
2. 輸出電壓 :  5~150V 單檔 10 轉調整  
 5~150V or 10~300V 兩檔 10 轉調整  
 5~150V or 10~300V or 20~600V 三檔 10 轉調整
3. 輸出電壓溫度係數 :  $\pm 0.2\%$  每攝氏度
4. 輸出總諧波失真 :  $\leq 0.2\%$  (at 50/60Hz )(0.1%/100Hz)  
(THD-R 量測 80KHz BW)
5. 輸出相線 : 1 $\emptyset$  2W
6. 輸出容量 :  500VA  1KVA  2KVA  3KVA
7. 最大輸出 : 120V/4.2A(500VA) ; 120V/8.3A(1KVA)  
120V/16.7A(2KVA) ; 120V/25A(3KVA){at ACV $\pm 5\%$ }
8. 電源穩定率 :  $\pm 0.2\%$  (ACV $\pm 10\%$ )
9. 負載穩定率 :  $\pm 0.2\%$ (50/60Hz ; 電阻性負載 ; PF=1)(0.5%/100Hz)(註 1)  
測試電壓  150V Range : 120V (10%~100%負載)  
 300V Range : 240V (10%~100%負載)  
 600V Range : 480V (10%~100%負載)
10. 輸出頻率 :  指撥 45.00Hz~550.0Hz  
 指撥 300.0Hz~500.0Hz (only CF-TMP 系列)
11. 頻率解析度 : 0.1Hz
12. 頻率準確度 :  $\pm 0.01\%$
13. 頻率穩定率 :  $\pm 0.05\text{Hz}$
14. 頻率溫度係數 : 50 PPM 每攝氏度

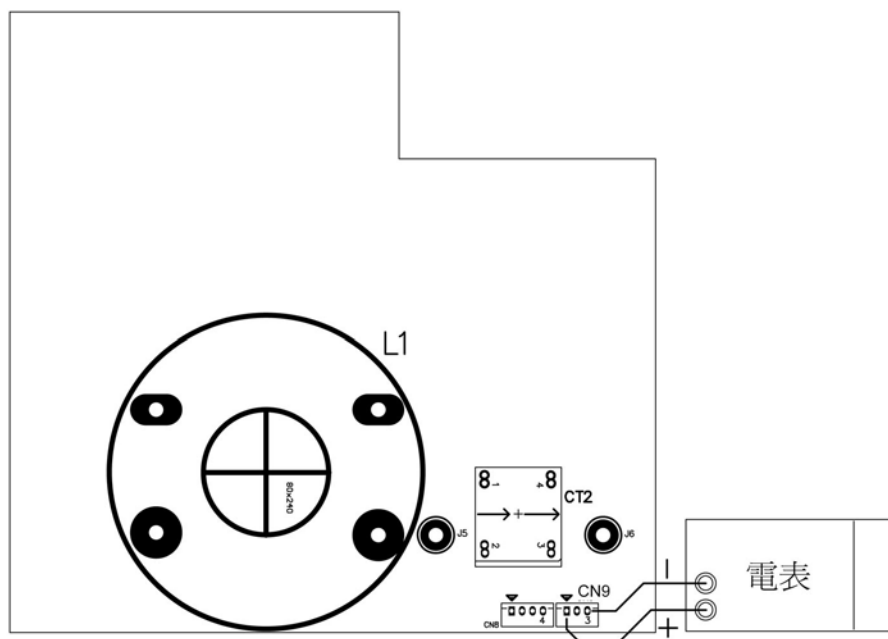
15. 頻率響應 :  $\pm 0.1\text{dB}$
16. 允許負載電流波形因素 : 3
17. 峰值電流 : 2.5 times the Max Current (RMS Value)
18. 負載功率因素 : 0.5 ~ 1 (lead or lag)
19. 反應速度 :  $100\ \mu\text{s}$
20. 電路結構 : Transistor Linear Amplifier Type (線性放大器)
21. 效率 : 40% (at full load)
22. 信號雜訊比 : More than 60db (at Rated Output Voltage)
23. 直流飄移 :  $< \pm 1\text{mV}$
24. 保護功能 : Over Load、Short、Over Temp (過載、短路、過溫度)
25. 冷卻系統 : 強制風冷
26. 操作環境 :  $0 \sim +45^{\circ}\text{C} / 10 \sim 90\% \text{RH}$  (不結露)
27. 絕緣電阻 :  $10\text{M}\Omega \ .500\text{VDC}$
28. 噪音 : 45~55dB 距離面板一米處
29. 指示表(四個視窗六個功能顯示) : 電壓、電流、頻率、瓦特、伏安、  
功因表

**註 1 :** 本機輸出電壓回授(反饋)補償點，位於換檔(檔位切換)電磁接觸或繼電器接點前，接點以後並未補償如需精確測量，請於補償點做負載穩定率測試(需開機蓋) (參考 FIG1-1~1-2)。

【FIG 1-1】 500T/1000T PCB 量測位置圖



【FIG1-2】 2KT/3KT PCB 量測位置圖



※ 以上規格如有變更，請參考網站上之規格

## 貳.配線說明

機種	輸入電源	輸入 最大電流	建議使用 無熔絲開關	輸入線徑	輸出線徑
CF-500T	1Ø2W 110V	12A	保險絲 10 A 2 支	14AWG	14AWG
	1Ø2W 220V	6A	保險絲 5 A 2 支		
CF-1000T	1Ø2W 110V	24A	保險絲 15A 2 支	14AWG	14AWG
	1Ø2W 220V	12A	保險絲 10 A 2 支		
CF-2000T	1Ø2W 220V	23A	30A	12AWG	12AWG
CF-3000T	1Ø2W 220V	33A	50A	10AWG	10AWG

註：(1)環境溫度：35℃以下。

(2)線材建議採用多芯絞線。

(3)配線時，導線應對絞。若導線超過 3 公尺時應再加粗一級，  
例：原 14AWG 改為 12AWG

(4)線材及無熔絲斷路器，請使用優良品牌。

(5)若輸出常需 0-150V 或 0-300V 切換使用時，導線應使用 0-150V  
設定之導線線徑。

(6)2000T 以上輸入電源固定為 AC 220V，如需改為 AC 110V 時，  
請另行訂製。

## 參.面板說明：(請參照 FIG 3-1)

- (1) POWER : 本機電源開關
- (2) OUTPUT : 輸出電源插座(每個插座最大 5A)
- (3) VOLTAGE : 輸出電壓調整器
- (4) RESET : 當 ALARM 動作時，作為重置功能
- (5) RANGE : 輸出電壓檔位選擇
- (6) OUTPUT : 輸出控制開關
- (7) FREQUENCY :  頻率 45~550.0Hz  
 頻率 300.0~500.0Hz
- (8) VLOTAGE DISPLAY : 數字電壓顯示表  
CURRENT DISPLAY : 數字電流顯示表
- (9) FUNCTION DISPLAY : 數字 PF、Hz、VA 顯示表
- (10) WATT DISPLAY : 數字瓦特顯示表
- (11) 量測功能選擇按鍵
- (12) 輸出高低電壓選擇開關
- (13) 高電壓設定半固定可調電阻
- (14) 低電壓設定半固定可調電阻
- (15) 狀態指示燈

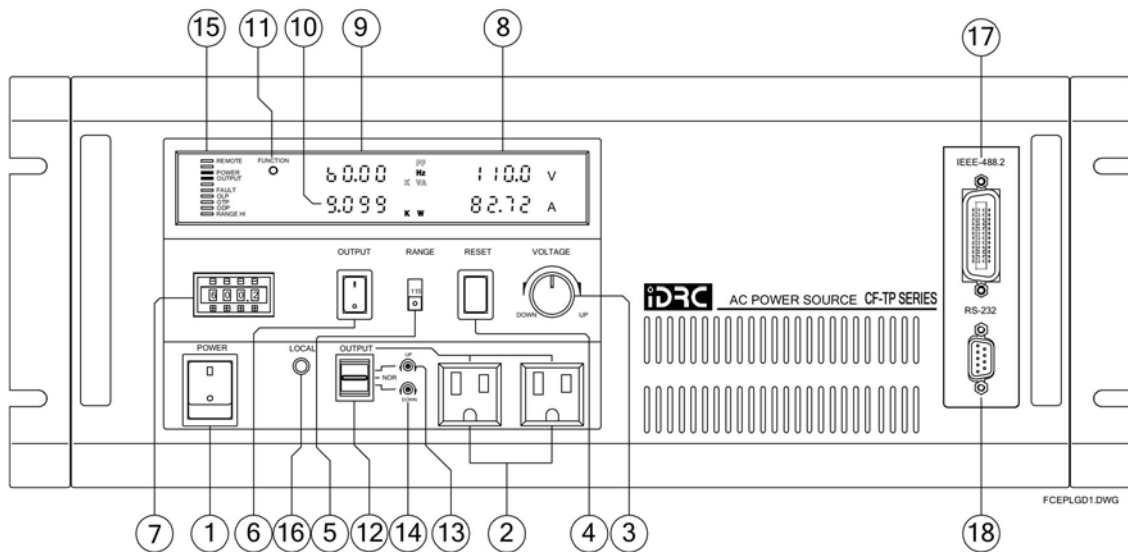
指示燈	功能說明	正常狀態	輸入電源異常	超載或短路	過溫度
REMOTE	電腦連線指示燈	ON/OFF ● / ○	●	●	●
POWER	電源輸入低於 20%或高於 20%指示燈	○	●	○	○
OUTPUT	輸出開關 ON/OFF 指示燈	ON/OFF ● / ○	○	○	○
FAULT	系統啟動保護模式警示燈	○	●	●	●
OLP	超載指示燈	○	○	●	○
OTP	晶體座溫度超過 90 度指示燈	○	○	○	●
ODP		○	○	○	○
RANGE HI	檔位開關指示燈	HI/LO ● / ○	○	○	○

備註:

- 表示指示燈亮，○表示指示燈不亮。
- 當有故障狀態時，輸出將會關閉，直到故障排除，重按 RESET 再繼續操作使用。

- (16) LOCAL : 設定為主機面板操作模式
- (17) IEE-488.2 : 與遠端個人電腦(PC)經由 GPIB 介面連接作雙向資料傳遞
- (18) RS-232 : 9 針 D 型母接頭,串接 RS232-C 通信連線與遠端個人電腦 (PC)作雙向資料及指令交換

【FIG 3-1】

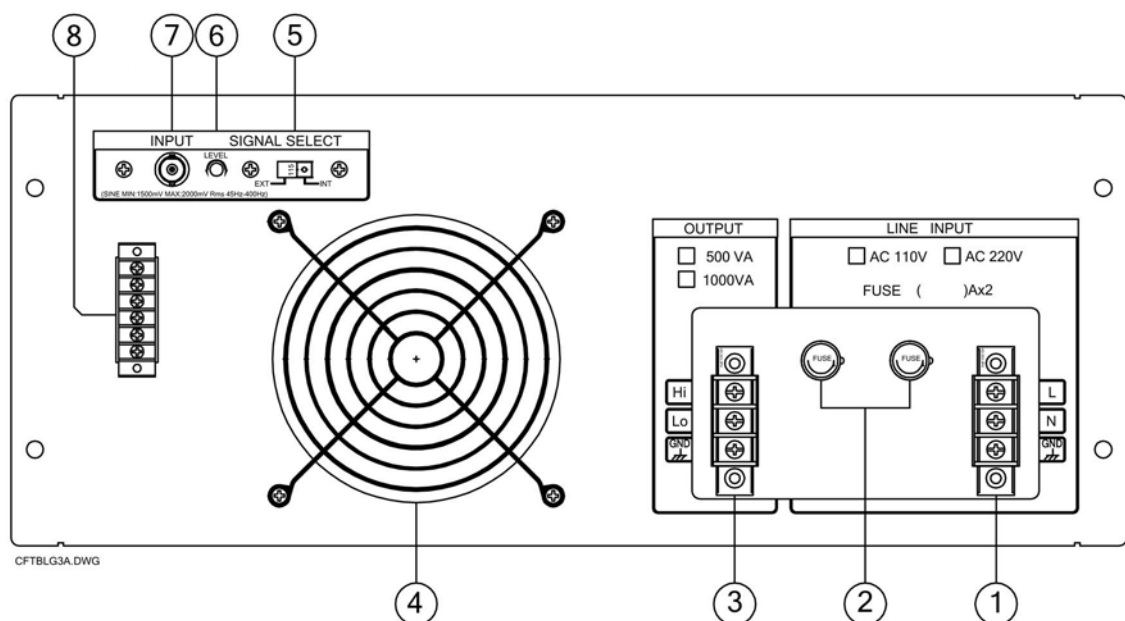




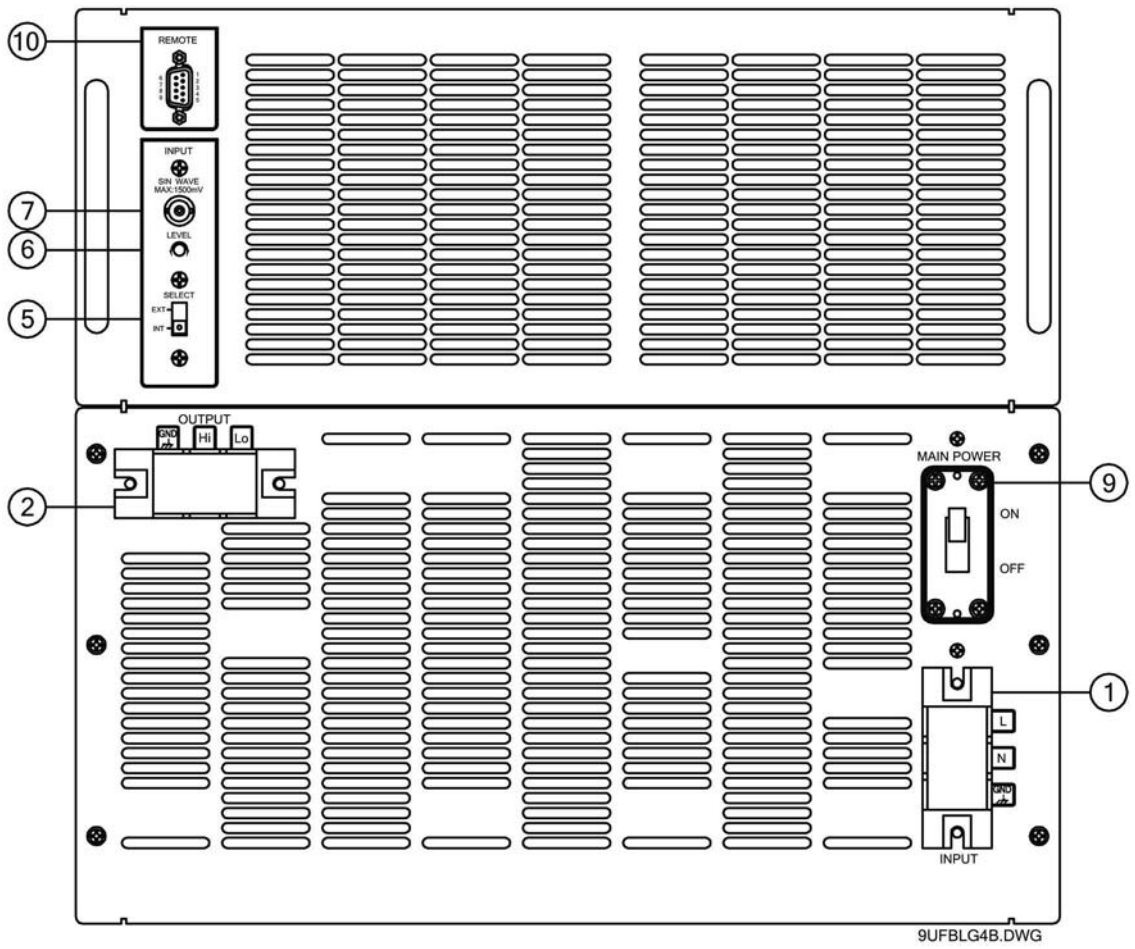
## 肆.背板說明：(請參照 FIG 4-1~FIG 4-2)

- (1) LINE INPUT : 本機輸入電源線座。
- (2) FUSE : 本機輸入電源保險絲。
- (3) OUTPUT : 變頻器輸出端子座。
- (4) FAN : 風扇通風口,請定期清除沾附之雜物及灰塵。
- (5) 外接或內部信號切換開關 (選購配備)。
- (6) 外接信號產生器微調旋鈕 (選購配備)  
: 當外部訊號過大時可由此微調旋鈕調整至適當大小或當接成三相時若三相出電壓有些許差異,可由此旋鈕調整之。
- (7) 外接信號產生器 BNC 輸入接頭 (選購配備)  
: 輸入阻抗為  $10K\Omega$ ,輸入信號建議為 Max 2000mV rms,頻率為 45Hz~550Hz 正弦波。
- (8) 外接控制端子座: 當有外部連線控制時才需用到此端子座。(選購配備)
- (9) 本機電源無熔絲開關
- (10) 遠端控制插座 (三相時使用)  
: 9PIN D 型接頭(請勿與 RS232 混用)

【FIG 4-1】



【FIG 4-2】



# 伍.操作程序

## 一．一般操作程序：

1. 變頻器背面通風處，應離物品 20CM 以上。
2. 將所有開關置於 OFF 狀態。
3. 將 VOLTAGE 調至最小。
4. 確認本機輸入電源無誤後，接上電源。
5. 打開本機輸入電源開關。
6. 送上電源後，須等 2~3 秒，本機即處於工作狀態。
7. 選擇所需頻率。
8. 選擇電壓檔位，若大於 150V 請選擇 300V 檔。
9. 調整所需電壓。
10. 接上負載後，再將輸出控制開關打開。

## 二．高低電壓設定操作程序：

將輸出高低電壓選擇開關切至 NOR，調整輸出電壓調整器（VOLTAGE）達所需的電壓值（如 110V），再將高低電壓選擇開關切至 UP。用調棒調高電壓半固定可調電阻，至所需電壓上升率（如：+10%，121V），再切至 Down。用調棒調低電壓半固定可調電阻，至所需電壓下降率（如：-10%,99V），最大可調範圍為±20%。

## ※ 注意事項：

1. 操作時若警報聲持續不停,請檢查負載是否過載或發生短路;檢查無誤之後再按 RESET。
2. 在打開輸入電源開關前，應先解除變頻器之負載，以免產生錯誤動作。

# 陸.GPIB、RS232 界面指令説明

## 1. IEEE488.2 Interface

- Specification : Standard IEE488.2

- Function :

- 1.SH1: Full Source Handshake
- 2.Ah1: Full Acceptor Handshake
- 3.T6: Basic Talker
- 4.L4: Basic Listener
- 5.SR0: Without Service Request
- 6.RL1: Remote/Local Change
- 7.PR0: Without Parallel Polling
- 8.DC1: Device Clear
- 9.DT0: Without Device Trigger
- 10.C0: Without Controlled function

- Command :

- \*CLS Clear the status byte summary register and all event registers .
- \*ESE <Enable Value>  
Enable bits in the standard Event status enable register.
- \*ESE? Query standard Event enable register.
- \*ESR? Query standard Event register.
- \*IDN Identification query.
- \*OPC Sets the “operation complete” bit(bit 0) in the Standard Event.
- \*OPC Return “1” to the output buffer after the command is executed.
- \*RST Reset the instrument to its power-on configuration.
- \*SRE <Enable Value>  
Enable bits in the Status Byte enable register.
- \*SRE? Query the Status Byte enable register.
- \*STB? Read status byte query.
- \*TST? Perform a complete self-test of instrument .  
Return “0” if the self-test is successful , or “1” if it test fails.

## 2.SCPI Command

- OUTPut

:OUTPut {ON/OFF/1/0}

:OUTPut:STATe {ON/OFF/1/0}

Sets the output of the AC source.

- SOURce

:SOURce:FREQuency <NRf>

Sets or query the output signal frequency.

:SOURce:RANGe {HIGH|LOW|AUTO}

Sets or query the output range.

:SOURce:SPPHase <NRf>

Sets or query the stop phase.

:SOURce:STPHase <NRf>

Sets or query Start phase.

:SOURce:TURN {ON/OFF/1/0}

Sets or Query the output signal status.

:SOURce:VOLTage <NRf>

Sets or query the output voltage.

- STATus

:PRESEnt

Clears the Questionable status register. both the event and enable registers are cleared.

:QUESTionable:CONDition?

Query the contents of condition register of Questionable status register group.

:QUESTionable:ENABLE

Sets or queries the enable register of Questionable status register group.

:QUESTionable[:EVENT]?

Query the contents of event register of Questionable status register group.

- SYSTem

:SYSTem:BEEPer

:SYSTem:BEEPer:IMMediate

:SYSTem:BEEPer:STATe {ON/OFF/1/0}

Sets the beeper to ON/OFF,queries the current setting.

:SYSTem:ERRor? (Query Only)

Queries the occurred error code and message.

:SYSTem:KLOCK

Sets or queries whether the front-panel keys are locked.

:SYSTEM:PRESet(No Query)

Resets to the default state.

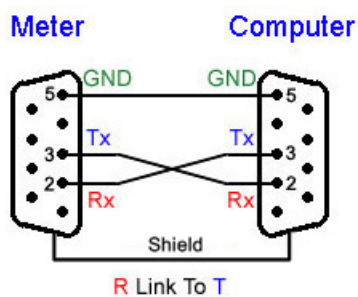
:SYSTEM:VERSion

Queries the value corresponding to the SCPI version to which the instrument complies

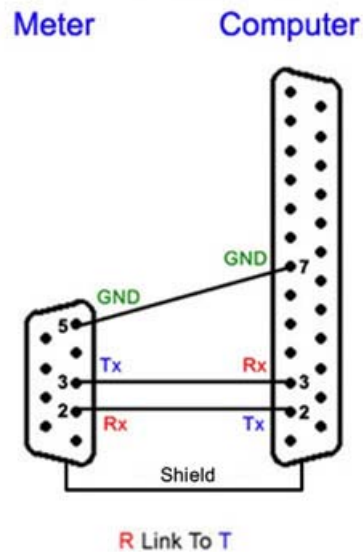
### 3.RS-232 Interface

- Mode : Start stop synchronization
- Baud rate : 1200.2400.4800.9600bps
- Command :
  - :SYSTEM:LOCal(No Query)  
Sets the system to local control.
  - :SYSTEM:REMote(No Query)  
Sets the system to remote control.
- RS-232 PC To CF/CIF AC SOURCE Connection

#### 9 PIN TO 9 PIN



#### 9 PIN TO 25 PIN



## 4.SCPI Status System

- Questionable Data

Event Register .or. Enable Register

Bit	Condition
10	FAULT :System Fault.
9	FUSE :Fuse Break.
8	ODP :Over Drive Protection.
3	OTP :Over Temperature Protection.
1	OLP :Over Load Protection. • Standard Event

Event Register .or. Enable Register

Bit	Condition
7	Not used
6	Not used
5	Command Error
4	Not used
3	Not used
2	Query Error
1	Not used
0	Operation Complete

- Status Byte

Summary Register .or. Enable Register

Bit	Condition
7	Operation Data
6	Request Service
5	Standard Event
4	Message Available
3	Questionable Data
2	
1	
0	

# 柒.錯誤碼對照表

Code	Message	Code	Message
0	No Error		
-100	Command error	-330	Self-test failed
-101	Invalid character	-350	Queue overflow
-102	Syntax error	-400	Query errors
-103	Invalid separator	-410	Query INTERRUPTED
-104	Data type error	-420	Query UNTERMINATED
-105	GET not allowed	-430	Query DEADLOCKED
-108	Parameter not allowed	-440	Query UNTERMINATED after indefinite response
-109	Missing parameter		
-112	Program mnemonic too long		
-113	Undefined header	60	ROM FAILED"
-121	Invalid character in number	61	RAM FAILED"
-123	Exponent too large	62	EEPROM R/W FAILED"
-124	too many digits	63	USER SETTING LOST"
-128	Numeric data not allowed	64	Command allowed only with RS-232
-131	Invalid suffix	65	Command not allowed in local
-138	Suffix not allowed	100	FAULT
-140	Character data error	101	OLP
-141	Invalid character data	102	OTP
-144	Character data too long	103	ODP
-148	Character data not allowed	104	FUSE
-150	String data error		
-151	Invalid string data		
-158	String data not allowed		
-160	Block data error		
-161	Invalid block data		
-168	Block data not allowed		
-170	Expression error		
-171	Invalid expression		
-178	Expression data not allowed		
-200	Execution errors		
-211	Trigger ignored		
-213	Init ignored		
-221	Setting conflict		
-222	Data out of rang		
-223	Too much data		
-224	Illegal parameter valve		
-230	Data corrupt or stale		
-241	Hardware missing		
-310	System error		
-311	Memory error		
-313	Calibration memory lost		



# 捌.RS232 鮑率與 GPIB 位址設定法

- 1.關閉電源開關(POWER)
- 2.頻率指撥開關撥於 0069
- 3.按著 RESET 開關同時開啟電源開關(POWER)
- 4.以上步驟若操作正確則 REMOT 燈亮，表示已進入設定模式
- 5.RS232 鮑率設定
  - a.指撥開關與鮑率關係如下：
    - 0100：1200
    - 0101：2400
    - 0102：4800
    - 0103：9600
  - b.依上列所示選定所需鮑率來設定指撥開關，如：9600 為 0103
  - c.按 RESET 鍵完成輸入動作
  - d.撥指撥開關於 0300 位置
  - e.按 RESET 鍵完成儲存動作，並跳出設定模式進入正常操作模式
  - f.將指撥開關撥於所需輸出電源頻率，如：50.00
- 6.GPIB 位址設定：
  - a.指撥開關與位址關係如下：
    - 0200：00
    - 0201：01
    - 0231：31
  - b.依上列所示選定所需位址，來設定指撥開關，如 0206 表示設定位址為 06
  - c.按 RESET 鍵完成輸入動作
  - d.調指撥開關於 0300 位置
  - e.按 RESET 鍵完成儲存動作，並跳出設定模式進入正常操作模式
  - f.將指撥開關設於所需輸出電源頻率，如：50.00

註：GPIB 內定 Address 為 06  
RS232 內定鮑率為 9600

# 玖.程式範例

## 範例 1 : GPIB demo Program

```
REM Example Program using NI-488 board level functions
REM QBDECL.BAS contains constants, declarations, and subroutine prototypes.
REM $INCLUDE: 'qbdecl.bas'
REM GPIBERR is an error subroutine that is called when a NI-488 function fails.
REM DVMERR is an error subroutine that is called when the IDRC CP-600 POWER
REM ANALYZER does not have valid data to send.
  DECLARE SUB gpiberr (msg$)
  DECLARE SUB dvmerr (msg$, rd$)
  CLS
  LOCATE 2, 3
  PRINT "CFxx Series demo Program ....."
  PRINT
  bdname$ = "GPIB0"
  CALL ibfind(bdname$, brd0%)
  IF brd0% < 0 THEN CALL gpiberr("Ibfind Error")
REM Send the Interface Clear (IFC) message.
  CALL ibsic(brd0%)
  IF (ibsta% AND EERR) THEN CALL gpiberr("Ibsic Error")
REM Turn on the Remote Enable (REN) signal.
  CALL ibsre(brd0%, 1)
  IF (ibsta% AND EERR) THEN CALL gpiberr("Ibsre Error")
REM Inhibit front panel control with the Local Lockout (LLO) command (hex 11).
REM Place the IDRC CP-600 in remote mode by addressing it to listen (ASCII "&").
REM Send the Device Clear (DCL) message to clear device (hex 14).
REM Address the GPIB interface board to talk (hex 40 or ASCII "@").
  cmd$ = CHR$(&H11) + "&" + CHR$(&H14) + "@"
  CALL ibcmd(brd0%, cmd$)
  IF (ibsta% AND EERR) THEN CALL gpiberr("Ibcmd Error")
REM set range to 150V
  wrt$ = ":SOUR:RANG LOW;"
  CALL ibwrt(brd0%, wrt$)
  IF (ibsta% AND EERR) THEN CALL gpiberr("Ibwrt Error")
REM set voltage to 100V
  wrt$ = ":SOUR:VOLT 100;"
  CALL ibwrt(brd0%, wrt$)
  IF (ibsta% AND EERR) THEN CALL gpiberr("Ibwrt Error")
REM set frequency to 60Hz
  wrt$ = ":SOUR:FREQ 60;"
  CALL ibwrt(brd0%, wrt$)
  IF (ibsta% AND EERR) THEN CALL gpiberr("Ibwrt Error")
REM set start phase
  wrt$ = ":SOUR:STPH 0;"
```

```

CALL ibwrt(brd0%, wrt$)
IF (ibsta% AND EERR) THEN CALL gpiberr("Ibwrt Error")
REM set stop phase
wrt$ = ":SOUR:SPPH 0;"
CALL ibwrt(brd0%, wrt$)
IF (ibsta% AND EERR) THEN CALL gpiberr("Ibwrt Error")
REM output relay on
wrt$ = ":OUTP ON;"
CALL ibwrt(brd0%, wrt$)
IF (ibsta% AND EERR) THEN CALL gpiberr("Ibwrt Error")
REM turn on signal
wrt$ = ":SOUR:TURN ON;"
CALL ibwrt(brd0%, wrt$)
IF (ibsta% AND EERR) THEN CALL gpiberr("Ibwrt Error")
SLEEP (3)
REM set voltage to 120V
wrt$ = ":SOUR:VOLT 120;"
CALL ibwrt(brd0%, wrt$)
IF (ibsta% AND EERR) THEN CALL gpiberr("Ibwrt Error")
SLEEP (3)
REM set frequency to 50Hz
wrt$ = ":SOUR:FREQ 50;"
CALL ibwrt(brd0%, wrt$)
IF (ibsta% AND EERR) THEN CALL gpiberr("Ibwrt Error")
SLEEP (3)
REM set voltage to 90V
wrt$ = ":SOUR:VOLT 90;"
CALL ibwrt(brd0%, wrt$)
IF (ibsta% AND EERR) THEN CALL gpiberr("Ibwrt Error")
SLEEP (3)
REM turn on signal
wrt$ = ":SOUR:TURN OFF;"
CALL ibwrt(brd0%, wrt$)
IF (ibsta% AND EERR) THEN CALL gpiberr("Ibwrt Error")
SLEEP (3)
REM Call the IBONL function to disable the hardware and software.
CALL ibonl(brd0%, 0)
END

SUB dvmerr (msg$, rd$) STATIC
PRINT msg$
PRINT "Status Byte = "; rd$
REM Call the IBONL function to disable the hardware and software.
CALL ibonl(brd0%, 0)
STOP
END SUB

SUB gpiberr (msg$) STATIC

```

```

PRINT msg$
PRINT "ibsta = &H"; HEX$(ibsta%); " <";
IF ibsta% AND EERR THEN PRINT " ERR";
IF ibsta% AND TIMO THEN PRINT " TIMO";
IF ibsta% AND EEND THEN PRINT " END";
IF ibsta% AND SRQI THEN PRINT " SRQI";
IF ibsta% AND RQS THEN PRINT " RQS";
IF ibsta% AND SPOLL THEN PRINT " SPOLL";
IF ibsta% AND EEVENT THEN PRINT " EVENT";
IF ibsta% AND CMPL THEN PRINT " CMPL";
IF ibsta% AND LOK THEN PRINT " LOK";
IF ibsta% AND RREM THEN PRINT " REM";
IF ibsta% AND CIC THEN PRINT " CIC";
IF ibsta% AND AATN THEN PRINT " ATN";
IF ibsta% AND TACS THEN PRINT " TACS";
IF ibsta% AND LACS THEN PRINT " LACS";
IF ibsta% AND DTAS THEN PRINT " DTAS";
IF ibsta% AND DCAS THEN PRINT " DCAS";
PRINT " >"
PRINT "iberr = "; iberr%;
IF iberr% = EDVR THEN PRINT " EDVR <DOS Error>"
IF iberr% = ECIC THEN PRINT " ECIC <Not CIC>"
IF iberr% = ENOL THEN PRINT " ENOL <No Listener>"
IF iberr% = EADR THEN PRINT " EADR <Address error>"
IF iberr% = EARG THEN PRINT " EARG <Invalid argument>"
IF iberr% = ESAC THEN PRINT " ESAC <Not Sys Ctrlr>"
IF iberr% = EABO THEN PRINT " EABO <Op. aborted>"
IF iberr% = ENEB THEN PRINT " ENEB <No GPIB board>"
IF iberr% = EOIP THEN PRINT " EOIP <Async I/O in prg>"
IF iberr% = ECAP THEN PRINT " ECAP <No capability>"
IF iberr% = EFSO THEN PRINT " EFSO <File sys. error>"
IF iberr% = EBUS THEN PRINT " EBUS <Command error>"
IF iberr% = ESTB THEN PRINT " ESTB <Status byte lost>"
IF iberr% = ESRQ THEN PRINT " ESRQ <SRQ stuck on>"
IF iberr% = ETAB THEN PRINT " ETAB <Table Overflow>"
PRINT "ibcnt = "; ibcnt%
REM Call the IBONL function to disable the hardware and software.
CALL ibonl(brd0%, 0)
STOP
END SUB

```

## 範例 2 : GPIB demo Program

```
#include <string.h>
#include "decl.h"

void gpiberr(char *msg);
void dvmerr(char *msg, char *rd);

char    read[512];          /* read data buffer          */
int     bd,                /* board or device number    */
        i;                /* FOR loop counter         */

void main() {

    clrscr();

    gotoxy(3,2);
    printf("CFxx Series demo Program .....");

    bd = ibfind ("GPIB0");
    if (bd < 0) gpiberr("ibfind Error");

/* Send the Interface Clear (IFC) message.    */

    ibsic (bd);
    if (ibsta & ERR) gpiberr("ibsic Error");

/* Turn on the Remote Enable (REN) signal.    */

    ibsre (bd,1);
    if (ibsta & ERR) gpiberr("ibsre Error");

/*
 * Inhibit front panel control with the Local Lockout (LLO) command
 * (hex 11). Place the IDRC CP-600 in remote mode by addressing it to listen
 * (hex 26 or ASCII "&"). Send the Device Clear (DCL) message to clear
 * internal device functions (hex 14). Address the GPIB interface board to
 * talk (hex 40 or ASCII "@").
 */

    ibcmd (bd,"\021&\024@",4L);
    if (ibsta & ERR) gpiberr("ibcmd Error");

/* set range to 150V          */
    ibwrt (bd,":SOUR:RANG LOW;", 15L);
    if (ibsta & ERR) gpiberr("ibwrt Error");
```

```

/* set voltage to 100V      */
ibwrt (bd,":SOUR:VOLT 100;", 15L);
if (ibsta & ERR) gpiberr("ibwrt Error");

/* set frequency to 60Hz   */
ibwrt (bd,":SOUR:FREQ 60;", 14L);
if (ibsta & ERR) gpiberr("ibwrt Error");

/* set start phase        */
ibwrt (bd,":SOUR:STPH 0;", 14L);
if (ibsta & ERR) gpiberr("ibwrt Error");

/* set stop phase         */
ibwrt (bd,":SOUR:SPPH 0;", 14L);
if (ibsta & ERR) gpiberr("ibwrt Error");

/* output relay on        */
ibwrt (bd,":OUTP ON;", 9L);
if (ibsta & ERR) gpiberr("ibwrt Error");

/* turn on signal         */
ibwrt (bd,":SOUR:TURN ON;", 14L);
if (ibsta & ERR) gpiberr("ibwrt Error");

sleep(3);
/* set voltage to 120V    */
ibwrt (bd,":SOUR:VOLT 120;", 15L);
if (ibsta & ERR) gpiberr("ibwrt Error");

sleep(3);
/* set frequency to 50Hz  */
ibwrt (bd,":SOUR:FREQ 50;", 14L);
if (ibsta & ERR) gpiberr("ibwrt Error");

sleep(3);
/* set voltage to 90V     */
ibwrt (bd,":SOUR:VOLT 90;", 14L);
if (ibsta & ERR) gpiberr("ibwrt Error");

sleep(3);
/* turn on signal         */
ibwrt (bd,":SOUR:TURN OFF;", 15L);
if (ibsta & ERR) gpiberr("ibwrt Error");

sleep(3);
/* Call the ibonl function to disable the hardware and software.      */

```

```

        ibonl (bd,0);

    }

void gpiberr(char *msg) {

    printf ("%s\n", msg);

    printf ("ibsta = &H%x  <", ibsta);
    if (ibsta & ERR ) printf (" ERR");
    if (ibsta & TIMO) printf (" TIMO");
    if (ibsta & END ) printf (" END");
    if (ibsta & SRQI) printf (" SRQI");
    if (ibsta & RQS ) printf (" RQS");
    if (ibsta & SPOLL) printf (" SPOLL");
    if (ibsta & EVENT) printf (" EVENT");
    if (ibsta & CMPL) printf (" CMPL");
    if (ibsta & LOK ) printf (" LOK");
    if (ibsta & REM ) printf (" REM");
    if (ibsta & CIC ) printf (" CIC");
    if (ibsta & ATN ) printf (" ATN");
    if (ibsta & TACS) printf (" TACS");
    if (ibsta & LACS) printf (" LACS");
    if (ibsta & DTAS) printf (" DTAS");
    if (ibsta & DCAS) printf (" DCAS");
    printf (" >\n");

    printf ("iberr = %d", iberr);
    if (iberr == EDVR) printf (" EDVR <DOS Error>\n");
    if (iberr == ECIC) printf (" ECIC <Not CIC>\n");
    if (iberr == ENOL) printf (" ENOL <No Listener>\n");
    if (iberr == EADR) printf (" EADR <Address error>\n");
    if (iberr == EARG) printf (" EARG <Invalid argument>\n");
    if (iberr == ESAC) printf (" ESAC <Not Sys Ctrlr>\n");
    if (iberr == EABO) printf (" EABO <Op. aborted>\n");
    if (iberr == ENEB) printf (" ENEB <No GPIB board>\n");
    if (iberr == EOIP) printf (" EOIP <Async I/O in prg>\n");
    if (iberr == ECAP) printf (" ECAP <No capability>\n");
    if (iberr == EFSO) printf (" EFSO <File sys. error>\n");
    if (iberr == EBUS) printf (" EBUS <Command error>\n");
    if (iberr == ESTB) printf (" ESTB <Status byte lost>\n");
    if (iberr == ESRQ) printf (" ESRQ <SRQ stuck on>\n");
    if (iberr == ETAB) printf (" ETAB <Table Overflow>\n");

    printf ("ibcnt = %d\n", ibcnt);
    printf ("\n");
}

```

```
/* Call the ibonl function to disable the hardware and software.      */  
  
    ibonl (bd,0);  
  
    exit(1);  
  
}  
  
void dvmerr(char *msg,char *rd) {  
  
    printf ("%s\n", msg);  
    printf("Status byte = %x\n", rd[0]);  
  
/* Call the ibonl function to disable the hardware and software.      */  
  
    ibonl (bd,0);  
  
    exit(1);  
}
```



### 範例 3 : RS232 demo Program

```
CLS
LOCATE 2, 3
PRINT "CFxx Series demo Program ....."
OPEN "COM1:9600,N,8,1,RS,CS0,DS0,CD0" FOR RANDOM AS #1
COM(1) ON

PRINT #1, " :SYST:REM;"           ' set device to remote mode
PRINT #1, " :SOUR:RANG LOW;"      ' set range to 150V
PRINT #1, " :SOUR:VOLT 100;"     ' set voltage to 100V
PRINT #1, " :SOUR:FREQ 60;"      ' set frequency to 60Hz
PRINT #1, " :SOUR:STPH  0;"      ' set start phase
PRINT #1, " :SOUR:SPPH  0;"      ' set stop phase
PRINT #1, " :OUTP ON;"           ' output relay on
PRINT #1, " :SOUR:TURN ON;"      ' turn on signal

SLEEP (3)
PRINT #1, " :SOUR:VOLT 120;"     ' set voltage to 120V

SLEEP (3)
PRINT #1, " :SOUR:FREQ 50;"      ' set frequency to 50Hz

SLEEP (3)
PRINT #1, " :SOUR:VOLT 90;"      ' set voltage to 90V

SLEEP (3)
PRINT #1, " :SOUR:TURN OFF;"     ' turn off signal

SLEEP (3)
PRINT #1, " :SYSTEM:LOC;"        ' set device to local mode
END
```

## 範例 4 : RS232 demo Program

```
#include "bios.h"
#include "conio.h"
#include "stdio.h"
#include "dos.h"

char err_no=0;

main()
{
    unsigned char readbuf[100],*p;
    int i;

    clrscr();
    gotoxy(3,2);
    printf("CFxx Series demo Program .....");

    initial_sys();          /* initial RS-232 */

    send(":SYSTEM:REM;");    /* set device to remote mode */

    send(":SOUR:RANG LOW;"); /* set range to 150V      */
    send(":SOUR:VOLT 100;"); /* set voltage to 100V   */
    send(":SOUR:FREQ 60;"); /* set frequency to 60Hz */
    send(":SOUR:STPH 0;");  /* set start phase      */
    send(":SOUR:SPPH 0;");  /* set stop phase       */
    send(":OUTP ON;");      /* output relay on      */
    send(":SOUR:TURN ON;"); /* turn on signal       */

    sleep(3);
    send(":SOUR:VOLT 120;"); /* set voltage to 120V   */

    sleep(3);
    send(":SOUR:FREQ 50;"); /* set frequency to 50Hz */

    sleep(3);
    send(":SOUR:VOLT 90;"); /* set voltage to 90V    */

    sleep(3);
    send(":SOUR:TURN OFF;"); /* turn off signal      */

    sleep(3);
    send(":SYSTEM:LOC;");   /* set device to local mode */
}
```

```

    }

    initial_sys()
    {
        outportb(0x3fb,0x80);
        outportb(0x3f8,0x0c);
        outportb(0x3f9,0x00);
        outportb(0x3fb,0x07);
        outportb(0x3fb,0x03);
    };

    send(unsigned char *p)
    {
        unsigned status;
        int    i,j,k;

        j=strlen(p);
        for(i=0;i<j;i++)
            {
                for(k=0;k<10000;k++)
                    {
                        status=inportb(0x3fd);
                        if(status & 0x20)
                            {
                                outportb(0x3f8,*p);
                                p++;
                                break;
                            }
                    }

                if(k==10000)
                    {
                        err_no=1;
                        gotoxy(3,22);
                        printf("RS232 sending timeout.....");
                        break;
                    }
            }
    }

    read(unsigned char *p)
    {
        unsigned char status,over;
        int    i,j;
        long int    k;

        over=0;

```

```

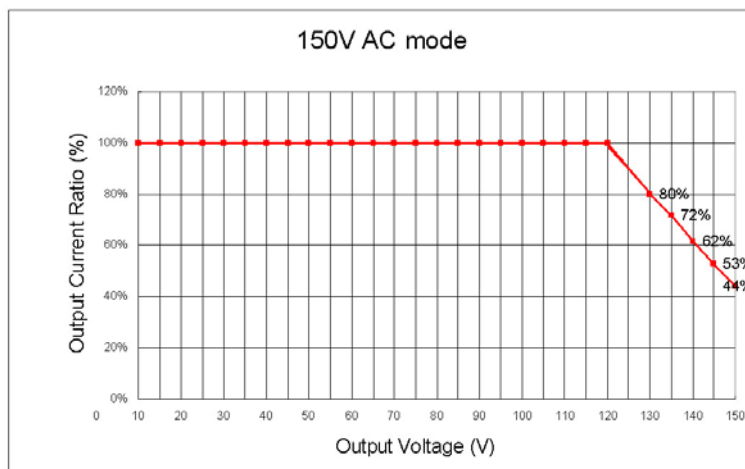
while(1)
{
    for(k=0;k<2000000;k++)
    {
        status=inportb(0x3fd);
        if(status & 0x01)
        {
            *p=inportb(0x3f8);
            if(*p==0x0a)
            {
                *++p=0;
                over=1;
            }

            p++;
            break;
        }
    }
    if(over==1)
        break;
    if(k==2000000)
    {
        err_no=1;
        gotoxy(3,22);
        printf("RS232 reading timeout.....\n");
        break;
    }
}
}

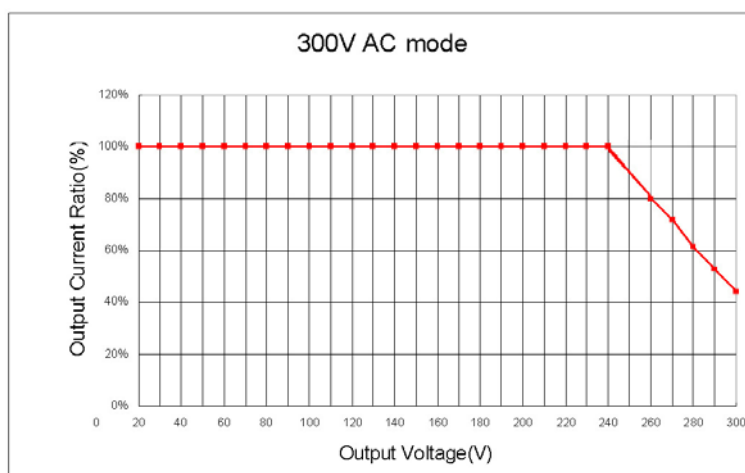
```

# 附錄壹：輸出特性曲線圖

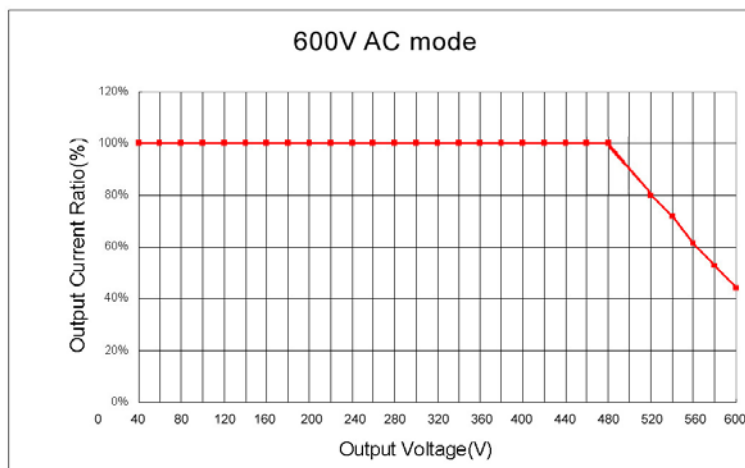
【F-G A1-1】 0~150V 檔位電壓/電流最大輸出特性曲線圖



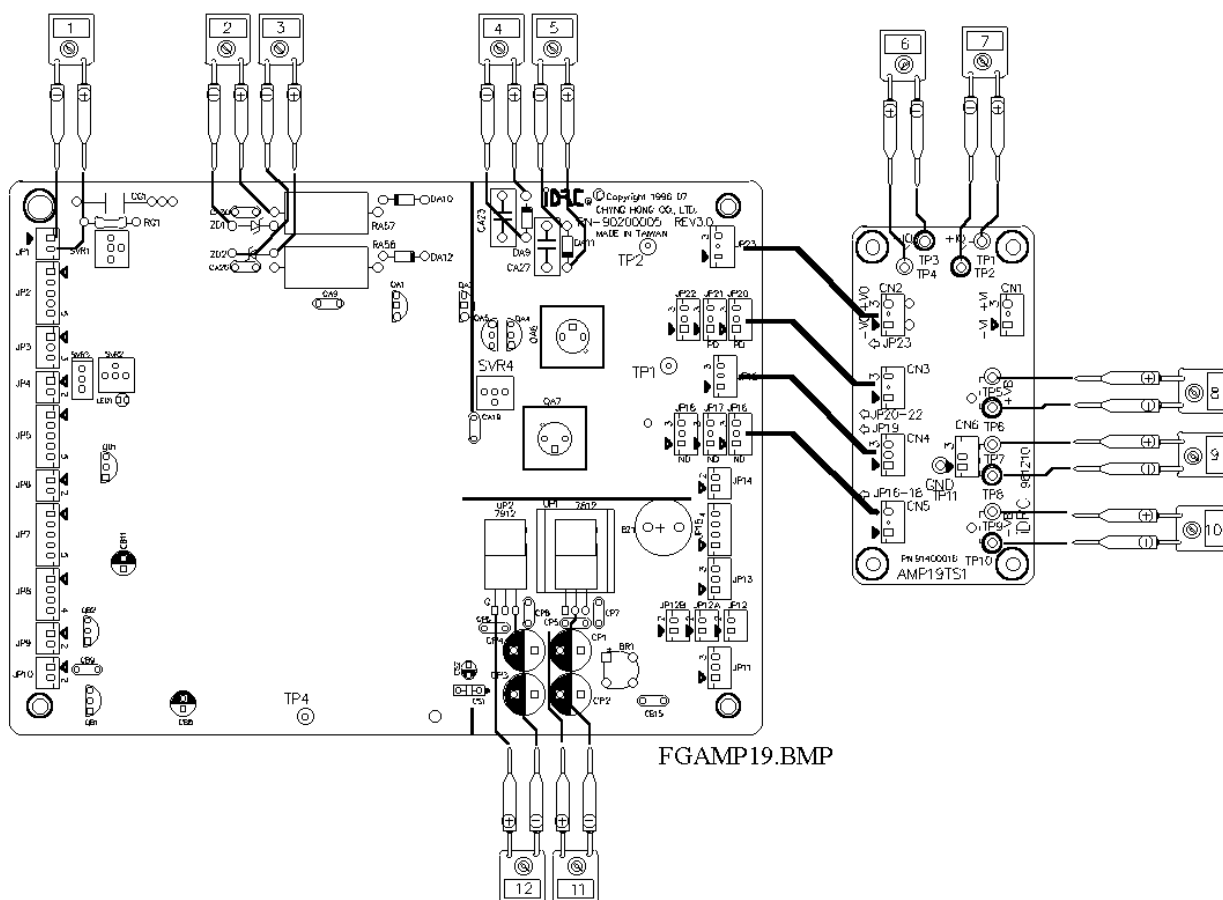
【F-G A1-2】 0~300V 檔位電壓/電流最大輸出特性曲線圖



【F-G A1-3】 0~600V 檔位電壓/電流最大輸出特性曲線圖



## 附錄貳：推動板故障檢修



### 一.連接測試治具(AMP19TS)步驟：

- 1.關閉本機電源。
- 2.拔掉 JP22,JP21,JP20,正半週推動連接線。
- 3.拔掉 JP18,JP17,JP16,負半週推動連接線。
- 4.拔掉 JP19 負回授連接線。
- 5.取測試治具(AMP19TS)及所附連接線依圖示連接。
- 6.再次檢視無誤後開啟電源,並將輸出電壓調器往左調到底。

## 二.量測點標準值：

### 1.電源部份：

測試錶別	測試值	備註
(2)	+15VDC±5%	OPAMP 正電源
(3)	-15VDC±5%	OPAMP 負電源
(4)	+45V±10%	推動級正電源
(5)	-45V±10%	推動級負電源
(12)	+12VDC±5%	控制部份正電源
(11)	-12VDC± 5%	控制部份負電源

### 2.信號部份：面板輸出電壓調整器往左調到底端。

測試錶別	測試值	備註
(6)(7)	±0.27V	推動板靜態工作電流
(8)(10)	±0.3~0.6V	後級晶體模組偏壓
(1)	0V	信號產生器輸入端
(9)	±2mV DC Max	推動板回授端

### 3.信號部份：面板輸出電壓調整器往右調到底端。

測試錶別	測試值	備註
(1)	AC 1500 mVRms±10%	信號產生器輸入端
(9)	AC 30VRms±10%	推動板負回授端

### 三.檢測步驟：

- 1.電源部份：參照步驟二之量測標準值無誤後，再進行下一步驟。如超出標準值時，則檢測相關零件 JP23，DA10，DA12，RA57，RA58，ZD1，ZD2。若仍無法解決時，請更換新板。
- 2.信號部份：當電錶(8)、(10)之量測值超出標準值時請調整 SVR4，若仍無法達成時請更換新板。當電錶(9)讀值大於 $\pm 2\text{mVDC}$ 時，請更換新板。  
**※注意：若電錶(9)讀值大於  $2\text{mVDC}$  時，切勿將晶體座接上，以免因輸出大量直流電而燒毀晶體座。**
- 3.信號部份：(面板輸出電壓調整器往右調到底)。當電錶(1)測試值正常，而電錶(9)測試值不正確時，請檢視 JP1、SVR1 等相關零件。若仍無法解決時，請更換新板。
- 4.以上量測點標準值若不符合測試範圍時，請勿接上晶體座送電，以免造成重大損害。



## 附錄參：導線線徑與電流規格表

※請注意！線材規格請依下列表格，方能正常使用。

使用 CD-SERIES 直流電源供應器或 CF、CIF 交流電源供應器時，需特別注意輸入與輸出導線之線徑問題，以防止因電流太大引起過熱，而造成意外，下列表格為導線在不同溫度下之線徑與電流規格表。

AWG NO	線徑 (約略值)	銅線溫度			
		60°C	75°C	85°C	90°C
		電流(A)			
14	2mm <sup>2</sup>	20	20	25	25
12	3.5mm <sup>2</sup>	25	25	30	30
10	5.5mm <sup>2</sup>	30	35	40	40
8	8mm <sup>2</sup>	40	50	55	55
6	14mm <sup>2</sup>	55	65	70	75
4	22mm <sup>2</sup>	70	85	95	95
3	30mm <sup>2</sup>	85	100	110	110
2	38mm <sup>2</sup>	95	115	125	130
1	50mm <sup>2</sup>	110	130	145	150
0	60mm <sup>2</sup>	125	150	165	170
00	70mm <sup>2</sup>	145	175	190	195
000	80mm <sup>2</sup>	165	200	215	225
0000	100mm <sup>2</sup>	195	230	250	260

導線的阻抗與其長度成正比與線徑成反比，並且直接影響 CD-SERIES 直流電源供應器或 CF、CIF 交流電源供應器的輸出特性；所以，往往在輸出端子上所量測出來的電壓不同於負載上的電壓，一般而言，這個電位差不得大於 0.5V(參考圖 A3-1)。備註：當電位差大於 0.5V 時，可將線徑加粗 1 倍或 2 倍甚至 3 倍。

下列表格為導線線徑每公尺所產生之阻抗規格表。

AWG NO	Resistance in $m\Omega/M$ (at 20°C)
22	52.8
20	33.5
18	20.96
16	13.19
14	8.30
12	5.22
10	3.277

FIG A3-1 (Cable length and voltage drop)

Condition : Voltage drop 0.5V

JIS C3307 IV Cable

