CF-AP 系列 交流電源供應器

操 作 手 冊

REVISION 1.0

FILE:CF-C-001

PN:800-01019

2001/6/5

目錄：

附錄：

壹.電氣規格

1. 輸入電壓　：AC 110V/220V ±10%　1Ø 2W 50/60Hz

2. 輸出電壓　：☐5～150V and 10～300V 兩檔 10 轉調整(Local Control)
☐1V~150V and 2V~300V 兩檔（電腦連線設定）

3. 輸出相線　：1Ø 2W

4. 電壓穩定率：±0.1% (ACV±10%)

5. 輸出電壓加載穩定率：±0.1% (註 1)

6. 輸出頻率　：☐指撥頻率：45Hz~550Hz (Local Control)
☐連線設定頻率：45Hz~550Hz (Remote Control)

7. 輸出容量　：　☐ CF-500AP(500VA)　☐ CF-1000AP(1KVA)
☐ CF-2000AP(2KVA)　☐ CF-3000AP(3KVA)
☐ CF-5000AP(5KVA)

8. 頻率響應　：≦0.2dB

9. 波形失真　：≦0.5% (註 1)

10. 溫度係數　：±0.01%／℃

11. 保護裝置　：Over Load、Short、Over Temp、AC 電源過高/過低保護

12. 散熱方式　：強制風冷

13. 指示表　　：數字電壓、電流、頻率、瓦特表、狀態顯示

註 1：電阻性負載(PF 大約爲 1)

貳.面板說明：(請參照 FIG 2-1)

(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，作爲重置功能
(5) RANGE ：輸出電壓檔位選擇
(6) OUTPUT ：輸出控制開關
(7) FREQUENCY ：45~550Hz
(8) VLOTAGE DISPLAY ：數字電壓顯示表
CURRENT DISPLAY ：數字電流顯示表
(9) FREQUENCY DISPLAY ：數字頻率顯示表
(10) WATT DISPLAY ：數字瓦特顯示表
(11)上限設定半固定可調電阻
(12)下限設定半固定可調電阻
(13)輸出上下限設定開關
(14) LOCAL ：設定爲主機面板操作模式
(15)狀態指示燈

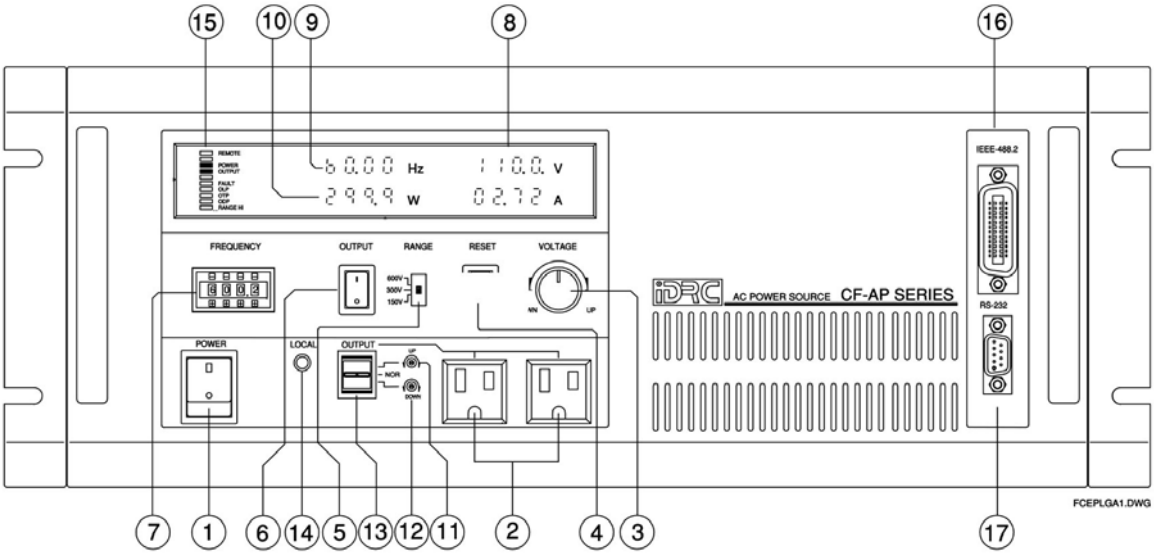
指示燈	功能說明	正常狀態	輸入電源異常	超載或短路	過溫度
REMOTE	電腦連線指示燈	ON/OFF ● / ○	●	●	●
POWER	電源輸入低於 20%或高於 20%指示燈	○	●	○	○
OUTPUT	輸出開關 ON/OFF 指示燈	ON/OFF ● / ○	○	○	○
FAULT	系統啓動保護模式警示燈	○	●	●	●
OLP	超載指示燈	○	○	●	○
OTP	晶體座溫度超過 90 度指示燈	○	○	○	●
ODP		○	○	○	○
RANGE HI	檔位開關指示燈	HI/LO ● / ○	○	○	○

備註:
1.●表示指示燈亮，○表示指示燈不亮。
2.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)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

(17)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

【FIG2-1】



參.背板說明：(請參照 FIG 3-1~FIG 3-4)

(1) LINE INPUT　：本機輸入電源端子座。

(2) FUSE　：本機輸入電源保險絲。

(3) LINE INPUT VOLTAGE SELECTOR

：本機輸入電源選擇,其接法如圖 3-1-1 所示。

(4) OUTPUT　：變頻器輸出端子座。

(5) RANGE 外部控制端子座(選購)

：輸入 DC 0V，0～150V(低檔)，

輸入 DC 12V，0～300V(高檔)

【注意：第(7)項開關切於 EXT 才有效。】

(6)外接控制端子座 ：當有外部連線控制時才需用到此端子座。

(7)外接或內部信號切換開關(須訂製)。

(8)外接信號產生器微調旋鈕(須訂製)

：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。

(9)外接信號產生器 BNC 輸入接頭 (須訂製)

：輸入阻抗爲 10KΩ,輸入信號建議爲

Max 2000mV Rms,頻率爲 45Hz～550Hz 正弦波。

(10) FAN　：風扇通風口,請定期清除沾附之雜物及灰塵。

(11)電源開關　：可恢復型電源過載保護開關。

(12)輸出開關　：輸出過載保護開關。

(13) PORT 1　：預留。

(14) PORT 2　：預留。

※注意：

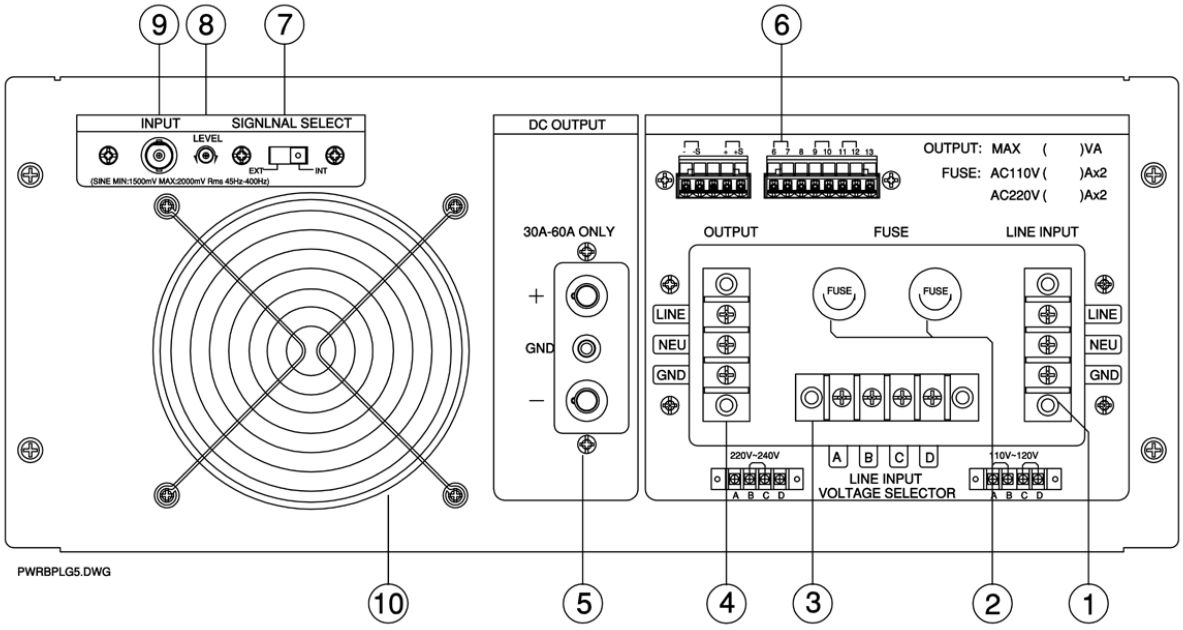
(1) 由 110V 換成 220V 時,只能 B、C 短路,千萬不可使 A、B 及 C、D 也短路。(如圖 3-1-1 所示)

(2) 輸入電源電壓設定:

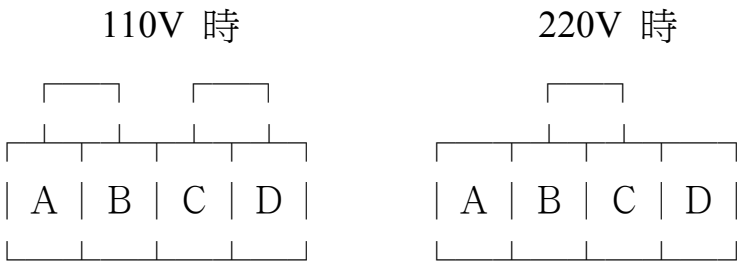
1. CF-500A 系列在後板(3)之端子座設定。
2. CF-1000A 須打開側板,由內部設定。
3. CF-2000A/CF-300A 輸入電壓 AC 220V,如須其他輸入電壓，敬請訂製。

(3) CF-500A/CF-1000A 輸入電壓出廠時一律接成 AC 220V。

【FIG 3-1】4U 背板



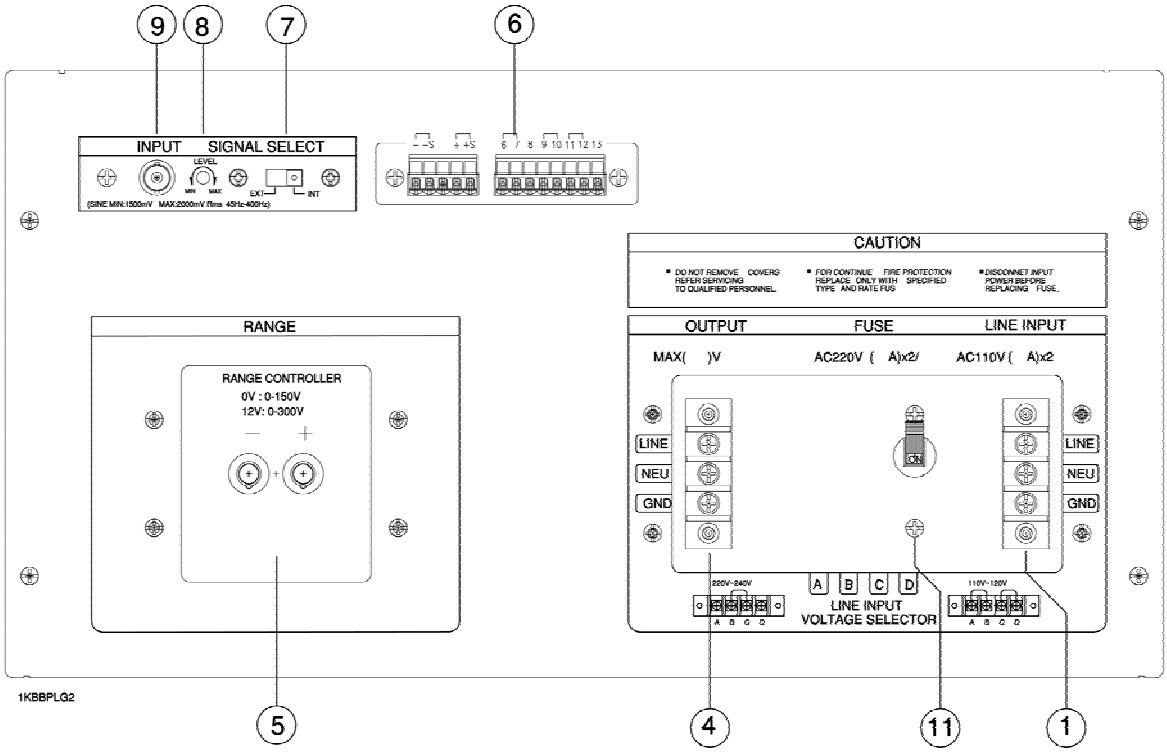
【FIG 3-1-1】



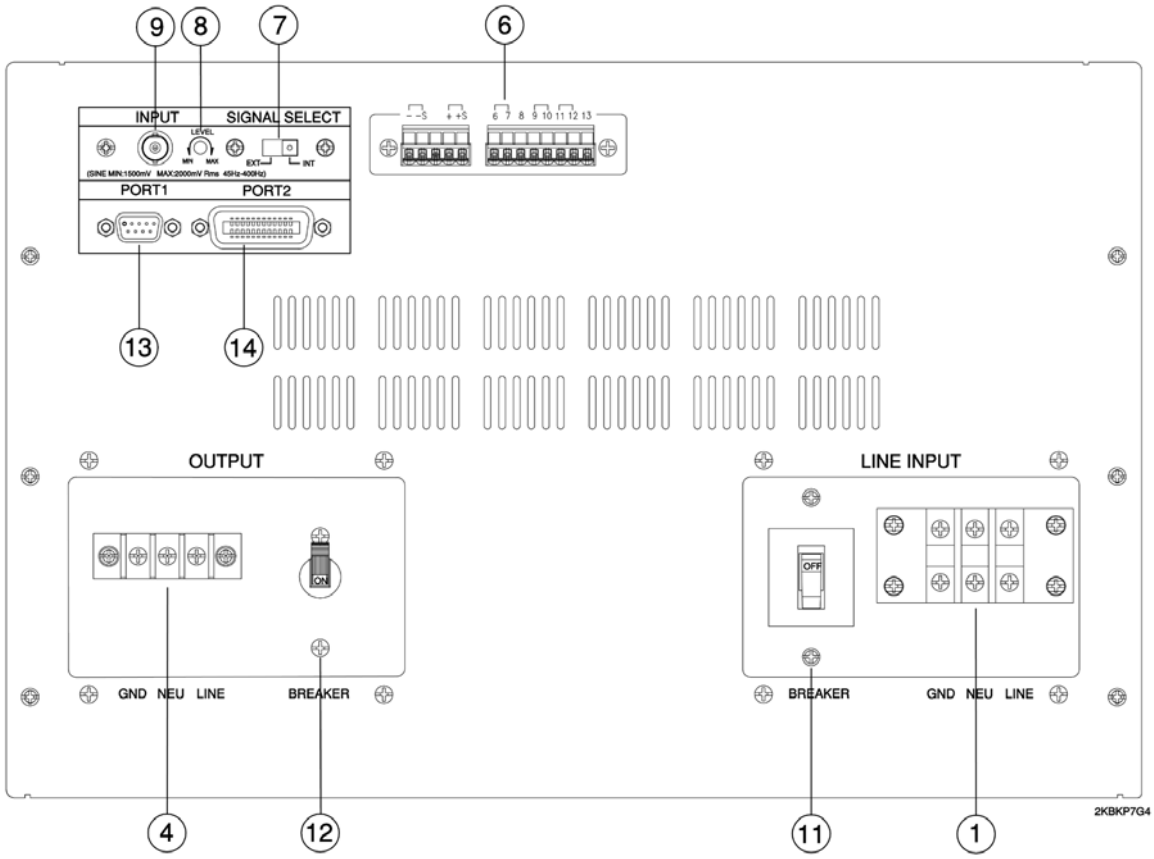
A.B 短路
C.D 短路

只有 B.C 短路
A.B 及 C.D 開路

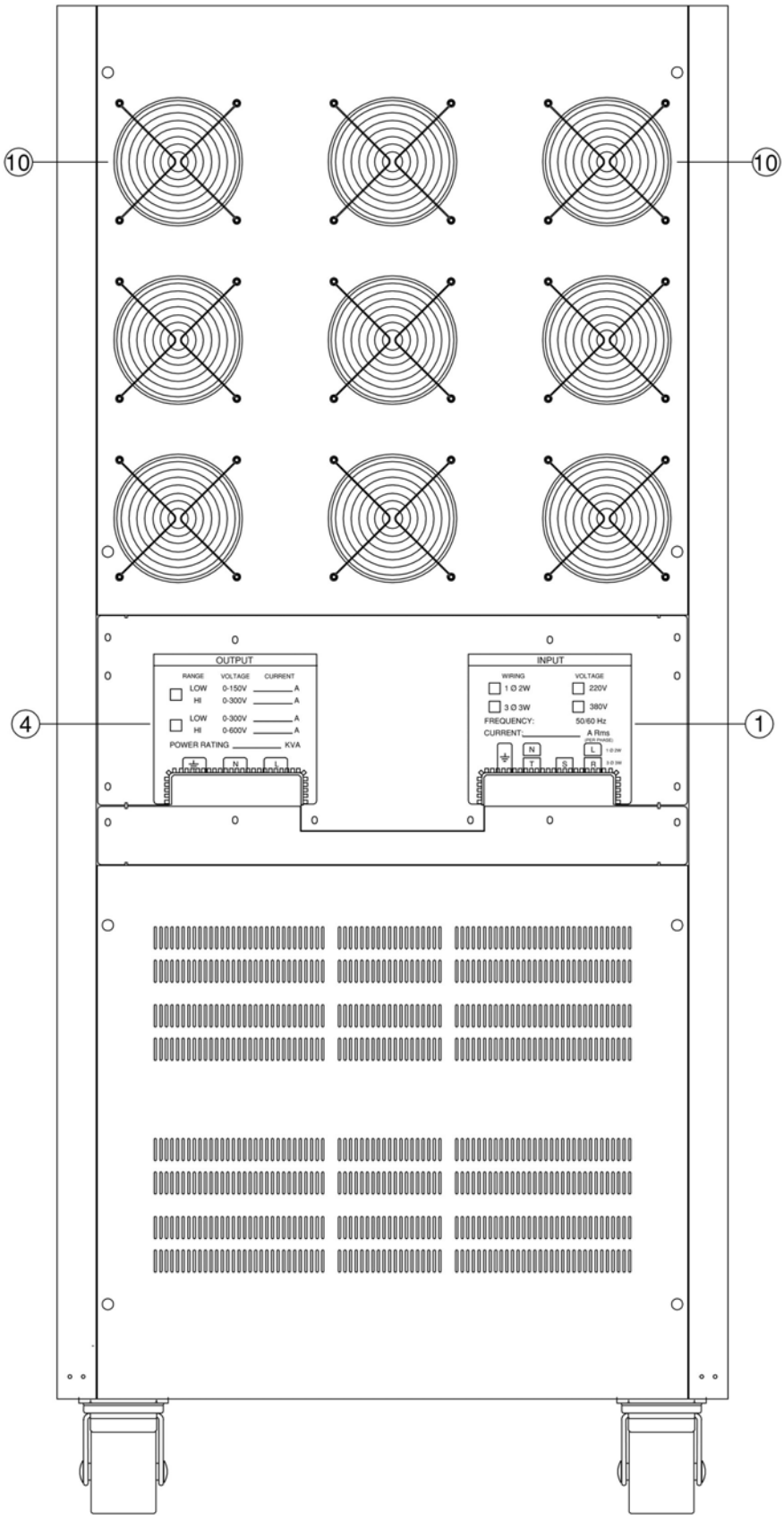
【FIG 3-2】9U 背板



【FIG 3-3】14U 背板



【FIG 3-4】28U 背板


10
10
4
1
OUTPUT
RANGE
VOLTAGE
CURRENT
LOW
HI
0-150V
0-300V
A
LOW
HI
0-300V
0-600V
A
POWER RATING
KVA
N
L
INPUT
WIRING
VOLTAGE
1 Ø 2W
220V
3 Ø 3W
380V
FREQUENCY:
50/60 Hz
CURRENT:
A Rms
N
L
T
S
R
28UAS07B

肆.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・上下限設定操作程序：

把輸出上下限設定開關切至 NOR，調整輸出電壓調整器(VOLTAGE) 達所需的電壓值（如 110V），再將輸出上下限設定開關切至 UP。用調棒調上限半固定可調電阻，至所需電壓上升率(如：+10%，121V), 再切至 Down。用調棒調下限半固定可調電阻，至所需電壓下降率（如：-10%,99V），最大可調範圍爲±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，以免產生誤動作。

伍.GPIB、RS232 界面指令說明

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 Reguest
  6.RL1: Remote/Local Change
  7.PR0: Without Parallel Polling
  8.DC1: Device Clear
  9.DT0: Without Device Trigger
  10.C0: Without Controllen 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 ebable 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 instrucment 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 instrucment complies

3.RS-232 Interface

- Mode : Start stop synchronization
- Baud rate : 1200.2400.4800.9600bps
- Command :

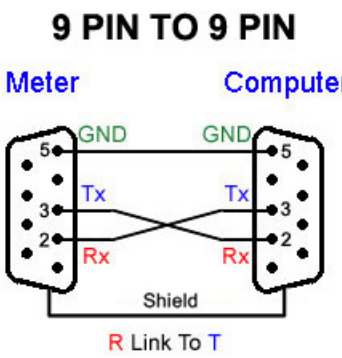
:SYSTem:LOCal(No Query)

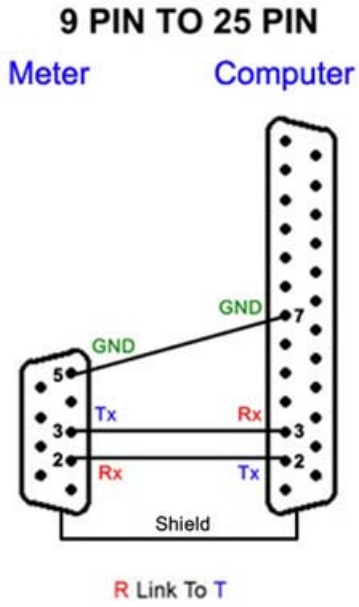
Sets the system to local control.

:SYSTem:REMote(No Query)

Sets the system to remote control.

- RS-232 PC To CF/CIF AC SOURCE Connection





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	Conditon
7	Operation Data
6	Request Service
5	Standard Event
4	Message Avilable
3	Questionable Data
2	
1	
0	

陸.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		
		60	ROM FAILED"
-113	Undefined header	61	RAM FAILED"
-121	Invalid character in number	62	EEPROM R/W FAILED"
-123	Exponent too large	63	USER SETTING LOST"
-124	too many digits	64	Command allowed only with RS-232
-128	Numeric data not allowed	65	Command not allowed in local
-131	Invaild suffix	100	FAULT
-138	Suffix not allowed	101	OLP
-140	Character data error	102	OTP
-141	Invalid character data	103	ODP
-144	Character data too long	104	FUSE
-148	Character data not allowed		
-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		

柒.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 開關同時開啓電源開關(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 爲 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 表示設定位址爲 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 爲 06

RS232 內定鮑率爲 9600

捌.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
```

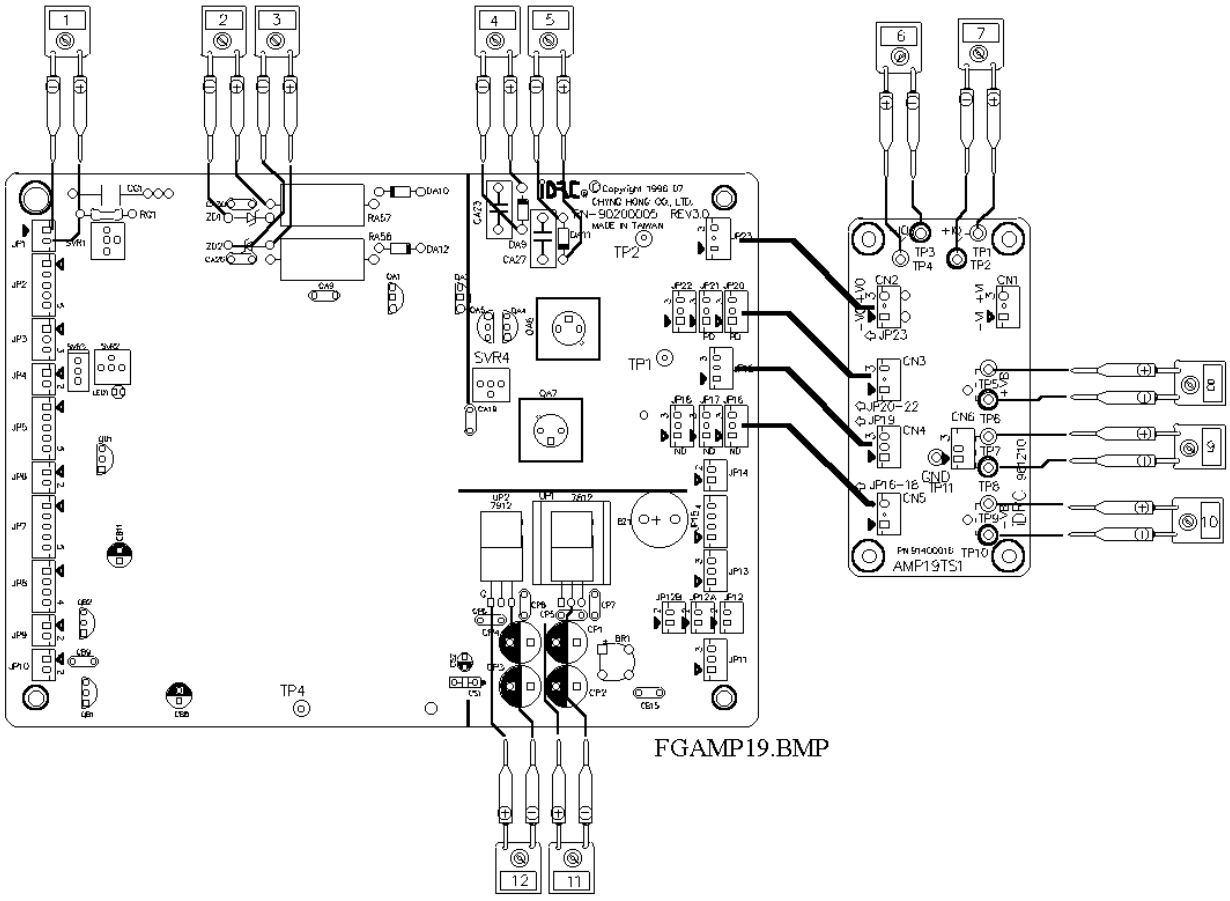
範例 4：RS232 demo Program

```
#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：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.再次檢視無誤後開啓電源,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。當錶(9)讀值大於±2mVDC時，請更換新板。

※注意：若錶(9)讀值大於 2mVDC 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)。當錶(1)測試值正常，而錶(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。

※注意：檢測時若無十足把握，不可輕易接上晶體座並送電，以免造成更大損害。

附錄貳：配線說明

機　型	輸入				輸出	
	電源	最大電流	建議使用無熔絲開關	建議使用線徑	最大電流 (Rms)	建議使用線徑
CF-500A 500VA	1Ø2W 110V	15A	15A	2.0mm²	120V 4.2A 240V 2.1A	2.0mm²
	1Ø2W 220V	7.5A	7.5A	2.0mm²		
CF-1000A 1KVA	1Ø2W 110V	30A	30A	5.5mm²	120V 8.4A 240V 4.2A	2.0mm²
	1Ø2W 220V	15A	15A	2.0mm²		
CF-2000A 2KVA	1Ø2W 220V	30A	30A	5.5mm²	120V 16.8A 240V 8.4A	2.0mm²
CF-3000A 3KVA	1Ø2W 220V	45A	50A	8mm²	120V 25.2A	3.5mm²
					240V 12.6A	2.0mm²
CF-5000A 5KVA	1Ø2W 220V	75A	75A	14mm²	120V 43A	8mm²
					240V 21.5A	3.5mm²

註：(1)線材建議採用多芯絞線。

(2)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 3.5mm²改爲 5.5mm²。

(3)N.F.B.：無熔絲斷路器。

(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)2000VA 以上輸入電源固定爲 AC 220V，如需改爲 AC 110V 時，請另行訂製。

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-A、CIF-A 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，下列表格爲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℃	75℃	85℃	90℃
		電流(A)			
14	2mm²	20	20	25	25
12	3.5mm²	25	25	30	30
10	5.5mm²	30	35	40	40
8	8mm²	40	50	55	55
6	14mm²	55	65	70	75
4	22mm²	70	85	95	95
3	30mm²	85	100	110	110
2	38mm²	95	115	125	130
1	50mm²	110	130	145	150
0	60mm²	125	150	165	170
00	70mm²	145	175	190	195
000	80mm²	165	200	215	225
0000	100mm²	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-A、CIF-A 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。**

下列表格爲導線線徑每公尺所產生之阻抗規格表。

AWG NO	Resistance in mΩ/M (at 20℃)
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