



第一章 系統管理 3

第一節 如何開啓 Agilent 3070 ? 3

第二節 如何關閉 Agilent 3070..... 5

第三節 UNIX 簡介..... 7

第二章 Agilent 3070 簡介..... 16

第一節 Agilent 3070 系統硬體簡介 16

第二節 Agilent 3070 系統軟體簡介 20

第三節 Agilent 3070 測試概念與流程簡介..... 23

第四節 BT-Basic 使用簡介 26

第三章 基本操作 31

第一節 如何載入治具及測試程式 31

第二節 如何解讀 error ticket..... 32

第三節 如何 debug analog 元件 38

第四節 如何 debug testjet..... 42



第五節 治具的維護與保養..... 46

第六節 如何使用 Operator Mode..... 49

第七節 常見之錯誤訊息..... 51

第四章 實驗..... 53

實驗一 Software Configure 設定..... 53

實驗二 使用線上手冊..... 56

實驗三 Find Pins 及 Verify Node 練習..... 58

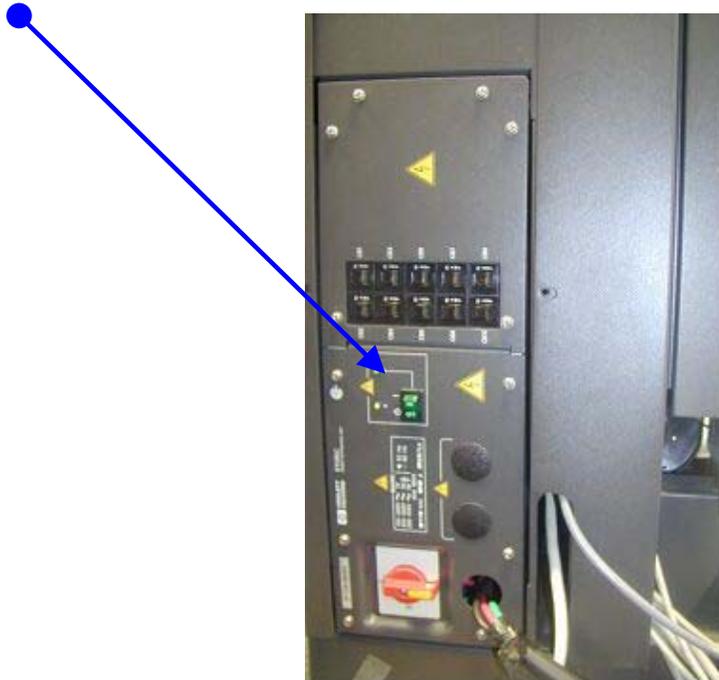


第一章 系統管理

第一節 如何開啓 Agilent 3070 ？

1. 打開空壓機的電源。
2. 打開 Agilent 3070 硬體的電源。(開關位置如下圖所示)

按下綠色按鈕



3. 打開工作站及螢幕的電源。



4.直到登入視窗出現後，輸入 **user1** 按下視窗中的 **OK** 按鈕或是按下鍵盤上的 **Enter** 鍵。



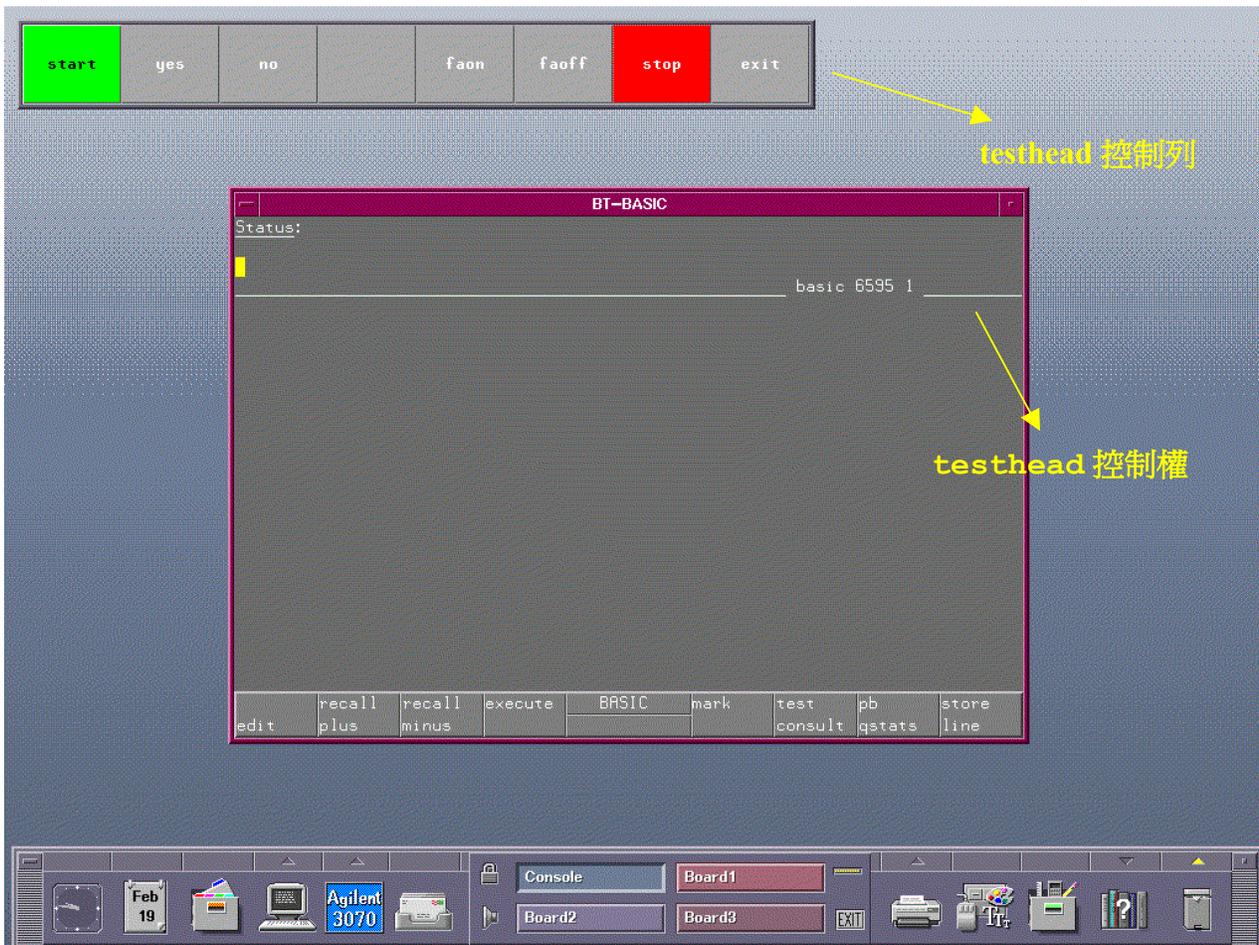
5.再輸入 password，按下視窗中的 **OK** 按鈕或是按下鍵盤上的 **Enter** 鍵。

6.利用滑鼠右鍵，開啓一個 BT-BASIC 視窗。

7.在 BT-BASIC 視窗的命令列上 (command line) 輸入：

testhead power on 再按下鍵盤上的 **Enter** 鍵

此時這個 BT-BASIC 視窗，在右上方屬性位置會多一個 " 1 "，視窗擁有 testhead 控制權，並出現 testhead 控制列，即可準備測試了。





第二節 如何關閉 Agilent 3070

有兩種方式可以關機：

方法 1 關機程序_一般程序

1. 在擁有 testhead 控制權的 BT-BASIC 輸入 **testhead power off**，按下鍵盤上的 **Enter** 鍵，當指令執行完畢，控制列消失後，即可關閉 testhead 電源。
2. 開啓一個 shell window，進入 super user mode。
3. 輸入 **cd /** 按 **Enter**，回到根目錄。
4. 輸入 **shutdown -h 0** 按 **Enter**。
5. 按 **N** 鍵。
6. 當畫面出現以下訊息時：

```
system has halted
OK to turn off power or reset system
UNLESS "WAIT for UPS to turn off power " message was printed
above
```

此時即可關閉工作站及螢幕電源。

方法 2 關機程序__利用 **shutdown user**

1. 在擁有控制權的 BT-BASIC 輸入 **testhead power off**，按下鍵盤上的 **Enter** 鍵，關閉 testhead 電源。
2. 按下 EXIT 的按鈕，離開 CDE 介面。
3. 在登入視窗輸入 **shutdown**，按下視窗上的 OK 按鈕。
4. 在登入視窗輸入 password，按下視窗上的 OK 按鈕。
5. 當畫面出現以下訊息時：

```
system has halted
OK to turn off power or reset system
UNLESS "WAIT for UPS to turn off power " message was printed
above
```

此時即可關閉工作站及螢幕電源。



補充：如何建立 **shutdown user** ? (optional)

1. 開啓一個 shell window，切換到 super user 模式，在提示符號 **#** 後輸入 **sam**。
2. 在 sam 的選單中，按下 Accounts for Users and Groups，再按下 Users。
3. 在新出現的視窗上按下 Actions => add。
4. 輸入使用者的名字 **shutdown** 以及 User ID (UID) 爲 **0**。
5. 按下 OK，再按下 yes (是否要重複使用 UID 0)，再按下 OK。
6. 出現 password 的視窗，可自行設定，不需密碼時，按下 Ok 即可。
7. 以 shutdown user 重新登入到系統中。
8. 用 BT-BASIC，開啓 **/home/shutdown/.dtprofile**，輸入以下兩行：

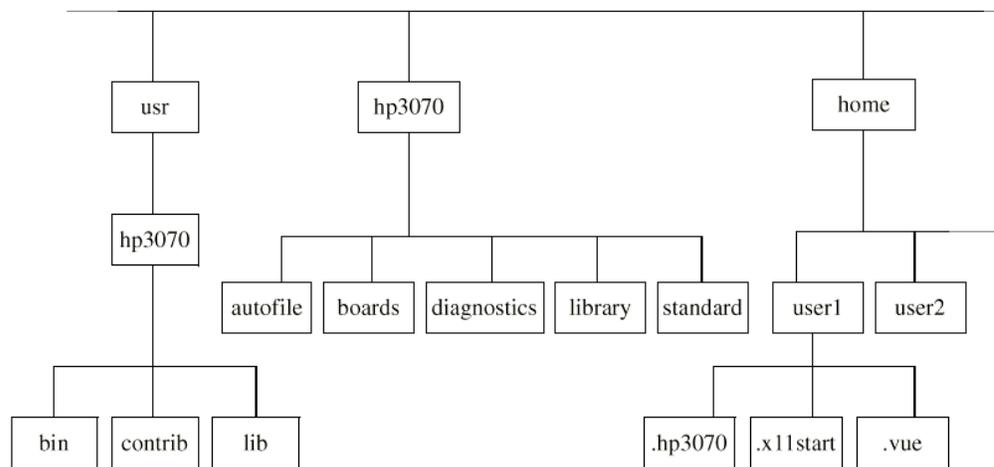
```
cd /  
  
/sbin/shutdwon -h -y 0
```

9. 存檔並關閉即可。



第三節 UNIX 簡介

HP-UX File structure



一、常用的UNIX指令：

Notes: 在UNIX環境中，英文字大小寫有差別。

↪ **cd** (Change Directory) 更改工作目錄。

↪ **pwd** (Present Working Directory) 列出目前的工作路徑。

範例.

```

$ cd /home/user1/class_bd    →到class_bd的路徑
$ pwd
/home/user1/class_bd
$ cd ..                    →回到上一層目錄
$ pwd
/home/user1
$ cd /                    →回到根目錄
$ pwd
/

```



↪ **bdf** 列出目前各個logic volume的使用情形。

範例.

```
$bdf
Filesystem           Kbytes  used   avail   %used   Mounted on
/dev/vg00/lvol4      1502173 116403 186038   88%     /home
```

↪ **ll** (List Long) 列出目前工作目錄下所有檔案的詳細資料。

例如：

```
d rwx rwx rwx  link number owner group size      date      name
      ↓   ↓   ↓
      owner group other
```

其中第一個英文字母代表檔案類型(file type)可以為：

- = ordinary 表示一般文件。
- d** = directory 表示這是一個目錄。
- l** = symbolic link 表示這個檔案會連結到某一個路徑。

其餘九個字母分別代表了檔案擁有者(owner)，同一個group的使用者，以及其他使用者有關Read、Write、Execute的權限，每一種權限分別以一個數字代表。

```
Read=4
Write=2
Execute=1
```

範例.

```
drwxrw-r-x 1 user1 hp3070 1024 Dec 17 21:00 class_bd
```

表示class_bd這個目錄，owner有rwx的權限，同一個group的user有rw的權限，其他的user有rx的權限。

↪ **ls** (List Short) 列出目前工作目錄下的檔案名稱（只有檔案名稱）。

↪ **lsf** (List Short, File of the directory specified) 列出目前工作目錄下的檔案名稱，並標記出可執行的檔案以及目錄。如：

```
fixture/                    '/' 表示這是一個目錄。
testplan                    表示這是一個檔案。
sorter*                     表示這是一個執行檔。
```

**範例.**

```

$ll
drwxrwxr-x  1  user1  hp3070  1024  Dec 17  21:00  class_bd
drwxr-x---  2  user1  hp3070  2048  Nov 12  15:03  Data
$ls
class_bd  Data
$lsf
class_bd\  Data

```

↪ **cp** (Copy Files) 複製檔案。

```

語法: cp source_file new_file      複製檔案
      cp -r dir new_dir            複製目錄
      cp source_file dest_dir      複製檔案到指定路徑

```

Note:

1. **絕對路徑**指以根目錄為參考點的路徑，例如“/home/user1/class_bd”，第一個“/”指的就是根目錄，這個路徑是指根目錄下的home下的user1下的class_bd子目錄。
2. **相對路徑**指的是以目前所在路徑為參考點的路徑，例如“class_bd/config”就是指目前路徑下的class_bd子目錄下的config檔。

1. 相對路徑的複製

範例.

```

$pwd
/hp3070/boards
$ls
data                                →目前/hp3070/boards下已經有一個檔案為data
$cp data data1                      →將data複製一個新的檔案名為data1
$ls
data data1                          →重新下ls 指令,此時/hp3070/boards下多了data1

```

範例.

```

$pwd
/hp3070/boards
$ls
class_bd
$cp -r class_bd class_bd1          →複製一個目錄
$ls
class_bd class_bd1                 →/hp3070/boards下多了class_bd1目錄

```



2. 絕對路徑的複製

範例.

```

$pwd
/hp3070/boards
$ls
class_bd
$ cp -r class_bd /home/user1/class_bd
                                →將class_bd目錄複製一份至/home/user1下

$ls
class_bd                        →此時/hp3070/boards下仍只有class_bd

$cd /home/user1                → 將路徑換至/home/user1下,發現多了
                                class_bd 目錄

$ls
class_bd

```

- ↳ **mv** (Move or Rename Files) 移動檔案或修改檔名。
 - 語法: `mv file file1` →將file更名為file1
 - `mv file dest_dir` →將file 移至其他路徑

範例.

```

$ls
data
$mv data test2                →將data更名為test2
$ls
test2
$mv test2 /home/user1/project/test2
                                →將test2移至/home/user1/project下

```

- ↳ **rm** (Remove Files) 移除檔案或目錄。
 - 語法: `rm file` →移除檔案
 - `rm -r dirname` →移除目錄

範例.

```

$ls
class_bd file
$ rm file
$rm -r class_bd

```



↳ **mkdir** (Create Directories) 新增目錄。

範例.

```
$mkdir test
```

↳ **rmdir** (Remove Directories) 移除空目錄。

範例.

```
$rmdir test
```

↳ **whoami** 列出目前使用者的身分

↳ **su** (Switch user) 更改使用者

語法: su <user ID> 切換到其他使用者

su 只輸入su, 表示切換到super user 模式

範例.

```
$su user2
```

password: →若user2有密碼, 請輸入user2 的密碼後再按Enter

```
$whoami
```

→列出目前的身分

```
user2
```

```
$su
```

password: →請輸入super user 的密碼後再按Enter

```
#
```

→出現#, 表示已經進入了super user mode

↳ **man** 使用線上手冊

語法 : man <command>

↳ **chmod** : 更改檔案的使用模式。

範例.

```
$ll
```

```
-rw-rw-rw 7 hp3070 1024 Aug 15 2001 class_bd
```

```
$ chmod 777 class_bd
```

```
$ll
```

```
-rwxrwxrwx 7 hp3070 1024 Aug 15 2001 class_bd
```



範例.

```
$ll
-rwxrwxrwx 7 hp3070 1024 Aug 15 2001 class_bd
$ chmod 776 class_bd
$ll
-rwxrwxrw- 7 hp3070 1024 Aug 15 2001 class_bd
```

範例.

```
$ll
-rwxrwxrw- 7 hp3070 1024 Aug 15 2001 class_bd
$ chmod 734 class_bd
$ll
-rwx-wxr-- 7 hp3070 1024 Aug 15 2001 class_bd
```

範例.

```
$ll
-rwx-wxr-- 7 hp3070 1024 Aug 15 2001 class_bd
$ chmod 764 class_bd
$ll
-rwxrw-r-- 7 hp3070 1024 Aug 15 2001 class_bd
```

↪ **exit** 結束關畢Shell 視窗

二、Changing Ownership

↪ 更改owner :

chown new_owner path/file

改變一個檔案的owner

chown -R new_owner path/folder

改變一個目錄, 及以下所有子目錄的owner



範例：

1.

```
$ll
-rwxrwxr-x 4 user1 hp3070 1024 Aug 15 2001 board
$chown user2 board
$ll
-rwxrwxr-x 4 user2 hp3070 1024 Aug 15 2001 board
```

2.

```
$ll
drwxrwxr-x 4 user1 hp3070 1024 Aug 15 2001 class_bd
$chown -R user2 class_bd
$ll
drwxrwxr-x 4 user2 hp3070 1024 Aug 15 2001 class_bd
```

↳ 更改group：

```
chgrp new_group path/file          改變一個檔案的group
chgrp -R new_group path/folder     改變一個目錄, 及以下所有子目錄的group
```

範例：

```
$ll
drwxrwxr-x 4 user1 hp3070 1024 Aug 15 2001 test1
$chgrp class2 test1
$ll
drwxrwxr-x 4 user1 class2 1024 Aug 15 2001 test1
```

↳ 同時更改owner和group：

```
chown new_owner:new_group path/file
```

範例：

```
$ll
drwxrwxr-x 4 user1 hp3070 1024 Aug 15 2001 data
$chown user2:class2 data
$ll
drwxrwxr-x 4 user2 class2 1024 Aug 15 2001 data
```



三、 Directory Backups

↩	tar cv	將程式tar到帶子中。
↩	tar xv	將程式由帶子tar到機器中。
↩	tar rv	將程式tar到帶子上，但是會接在帶子原有的程式後面。
↩	tar tv	列出帶子所tar的程式，但是不tar到機器上。
↩	tar cvf filename.tar filename	將檔案壓縮成.tar檔。
↩	tar xvf filename.tar	將.tar檔還原。
↩	compress filename.tar	將.tar 檔壓縮成.Z檔。
↩	uncompress filename.tar.Z	將.Z檔還原為.tar檔。

範例：

1. 如果在/home/user1/project下有test1、test2、test3三個目錄，要將其中test1 tar到磁帶中？
\$cd /home/user1/project
\$tar cv test1
2. 如果tar 完test1後還要將test2 接著用同一磁帶tar起來呢？
\$tar rv test2 記住這次用rv,如果再用cv 那裡磁帶中的test1會被覆蓋過去
3. 檢查資料是否都tar 好了：
\$tar tv
4. 將磁帶中的資料tar 回工作站的硬碟中：
\$cd /home/user1/project2 a先指定要放資料的錄徑
\$tar xv
5. 將test 壓縮成.tar檔：
\$tar cvf test.tar test
6. 將test.tar 還原為test：
\$tar xvf test.tar
7. 將test.tar 壓成.Z檔：
\$compress test.tar
8. 將test.tar.Z 還原為.tar：
\$uncompress test.tar.Z



補充:rmbkup 公用程式:清除 backup 檔案(~檔)

第一次使用 **rmbkup** 指令前請先執行以下指令

1. 進入 super user 模式
2. 更改路徑
#cd /opt/hp3070/contrib/bin
3. 執行以下程式
#sh Convert

範例:

- 1.清除/home/user1 下所有的 backup 檔案
\$cd /home/user1
\$rmbkup
- 2.清除根目錄下所有的 backup 檔案
\$cd /
\$rmbkup



第二章 Agilent 3070 簡介

第一節 Agilent 3070 系統硬體簡介

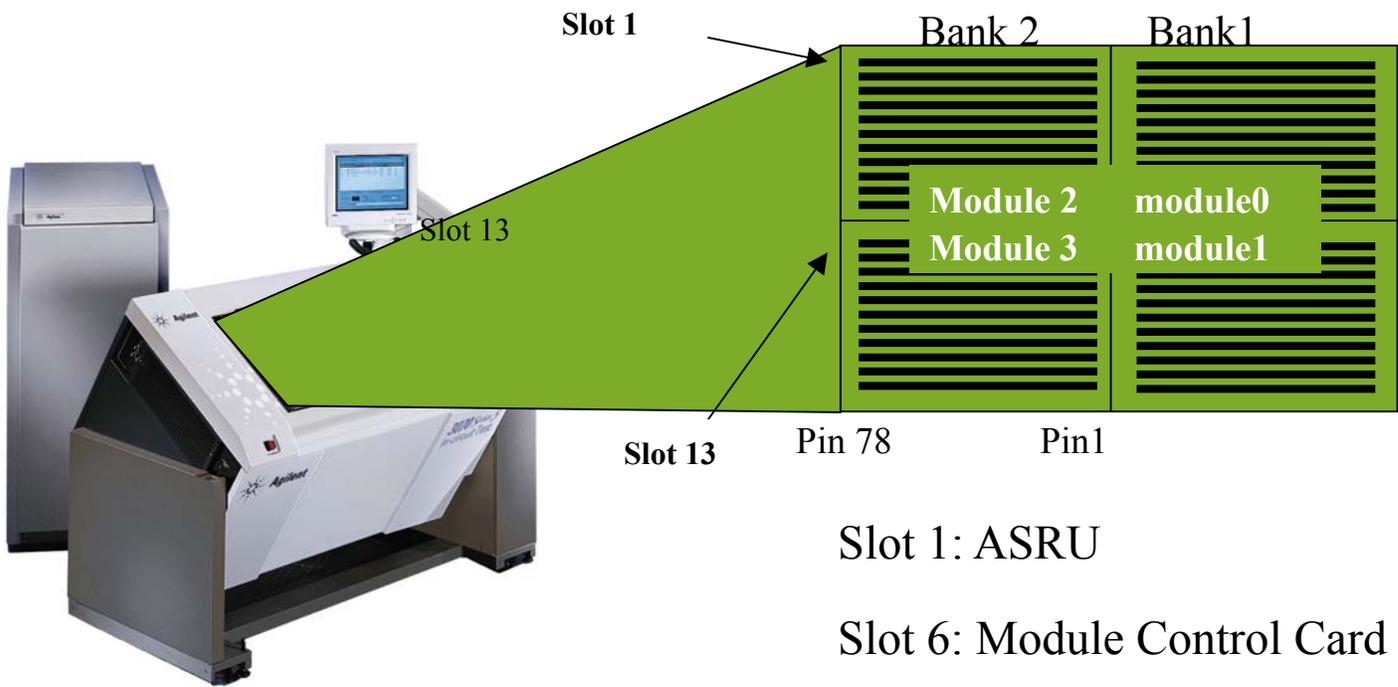
一、 簡介

3070 系統可以大致分為三個部分，分別為工作站 (Controller)，**testhead** 以及**擴充箱** (Support Bay)。

Note:擴充箱只有在 307x 的系統中才需要用到，而在 317x 及 327x 的系統中並不一定要使用。

二、 硬體架構





1. **Module and Bank**：依系統的型號不同，會有不同數量的 module。Module 和 bank 的定義如上圖所示。在 module 中會有許多的卡片，各有各的用途。以下是各種卡片詳細的介紹。
2. **Module 中的卡片**：每個 module 可以插 9 片卡片。第一槽是 ASRU card,第六槽是 Control card，其他的則是 Pin card。

I. Control Card：

在每一個 Module 必須要有一片，也只能有一片，而且要放在 Slot 6 的位置。簡單來說，Control Card 的工作就是控制 module 中其他的卡片，告訴它們要做什麼，要怎麼做。如：在類比量測(Analog Test)時，Control Card 會發出命令讓某些 pin card 上的 relay(繼電器)關閉以形成量測迴路,並啟動 ASRU card 上的儀器進行量測。

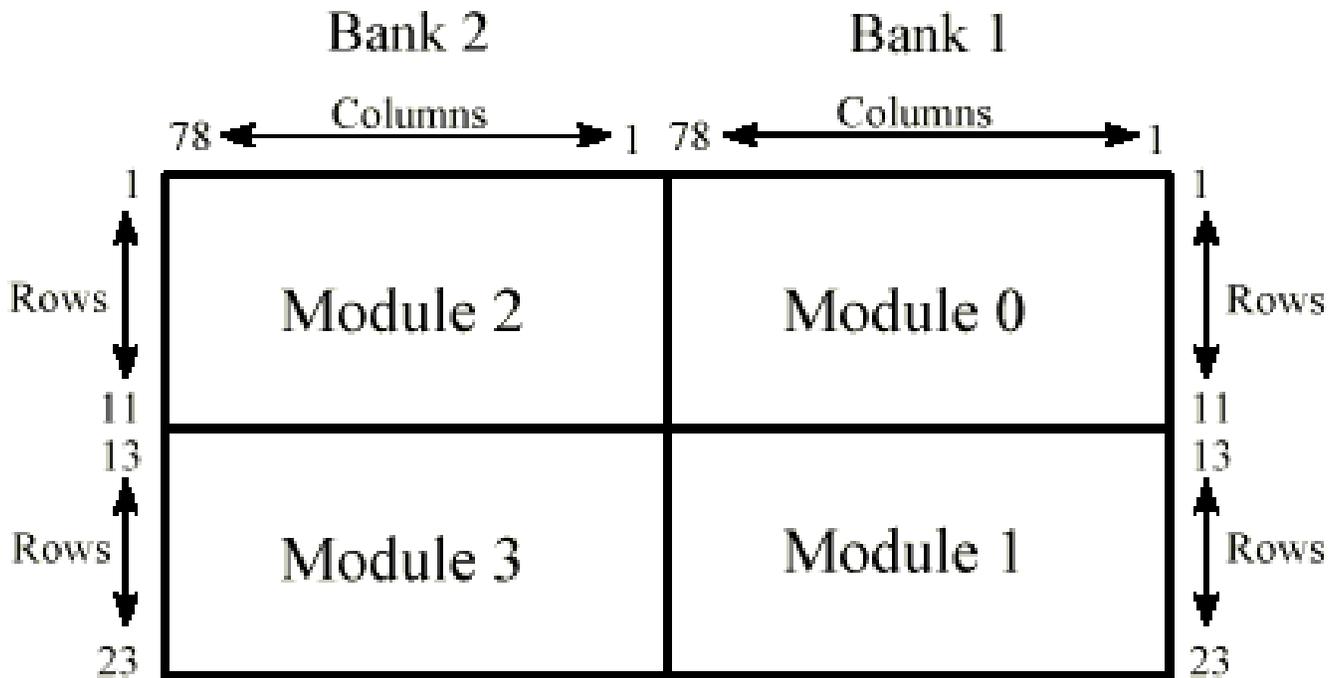
II. ASRU Card：

在每一個 Module 必須要有一片，也只能有一片，而且要放在 Slot 1 的位置。ASRU(Analog Stimulus/Response Unit) Card 的工作就是負責類比量測。在 ASRU Card 中有信號源(source)，偵測器(detector)以及一些用來做多工(multiplexing)的排線(bus)。



III. Pin Card :

在每一個 module 中至少要一片，而每一個 module 中可以放最多九片 Pin Card。Pin Card 上有 relay 來進行多工的動作,以形成量測迴路。Pin card 有兩種，hybrid 和 analog。Hybrid card 可提供數位和類比測試所需的線路，而 analog card 只有類比的線路。一個系統內 pin card 越多則可測的點數越高。



Front View of Testhead

User's Point of View (BRC)

3. BRC(Bank , Row , Column)的算法：在 3070 上有很多的卡片，而每一片卡片的每一支 Pin 的位置要怎麼來定義？我們用所謂的 BRC 來做定義(格式為 brcc)。所謂 B 就是 Bank，在圖中我們可以看到，從正面來看 3070 系統，右邊這兩個 module (module 0 及 module 1)稱為 Bank 1；而左邊這兩個 module (module 2 及 module 3)稱為 Bank 2。所謂 R 就是 Row，在圖中我們可以看到，Row 是從上到下來的，可是一個 module 至多只有 11 片卡片(包含一片 ASRU Card 及一片 Control Card)為何會有第 23 列呢？這是因為在 module 及 module 之間有一條分隔槽，而為了方便計算，我們把這一條分隔槽也算進去。所以 module 3 的第一列應該是 13 才對。至於 C 就是 Column，在每一片 Card 都有 78 支 Pin，從右至左算。

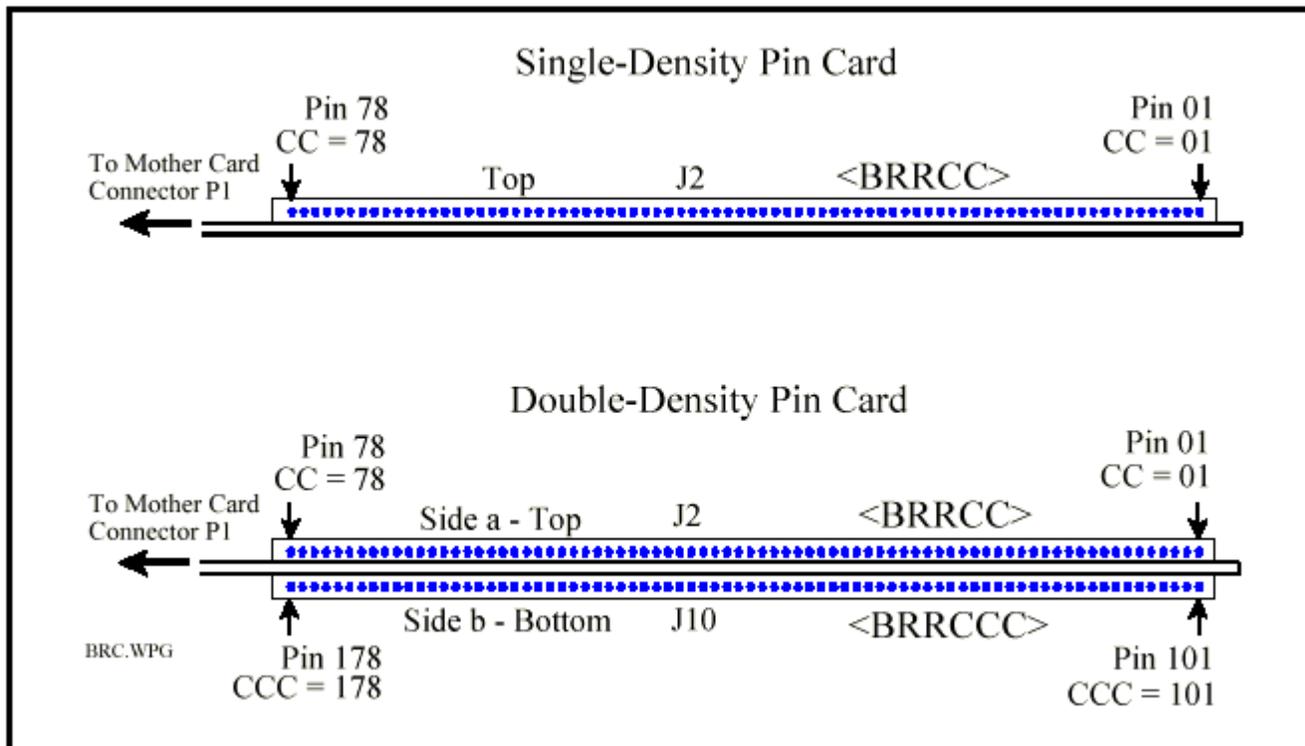


例如：

BRC=20841 代表是第 2 個 Bank (Bank 2)的第 8 列(module2 的第 8 列)的第 41 支 pin。

BRC=21355 代表是第 2 個 Bank (Bank 2)的第 13 列(即 module3 的第 1 列)的第 55 支 pin。

BRC=122167 我們可以很快的知道這是在 Bank 1，第 22 列。但是爲什麼在 67 行之前多了一個“1”？這是因爲有些卡片是有兩排(Double Density)的而有些是只有一排(Single Density)的。只要是在 Column 之前多了一個“1”，代表這是一片 Double Density 的卡片，並且現在的 BRC 指的就是這片卡片的第二排。所以這個 BRC 所指的是 Bank 1 第 22 列卡片的第 2 排的第 67 支 Pin。



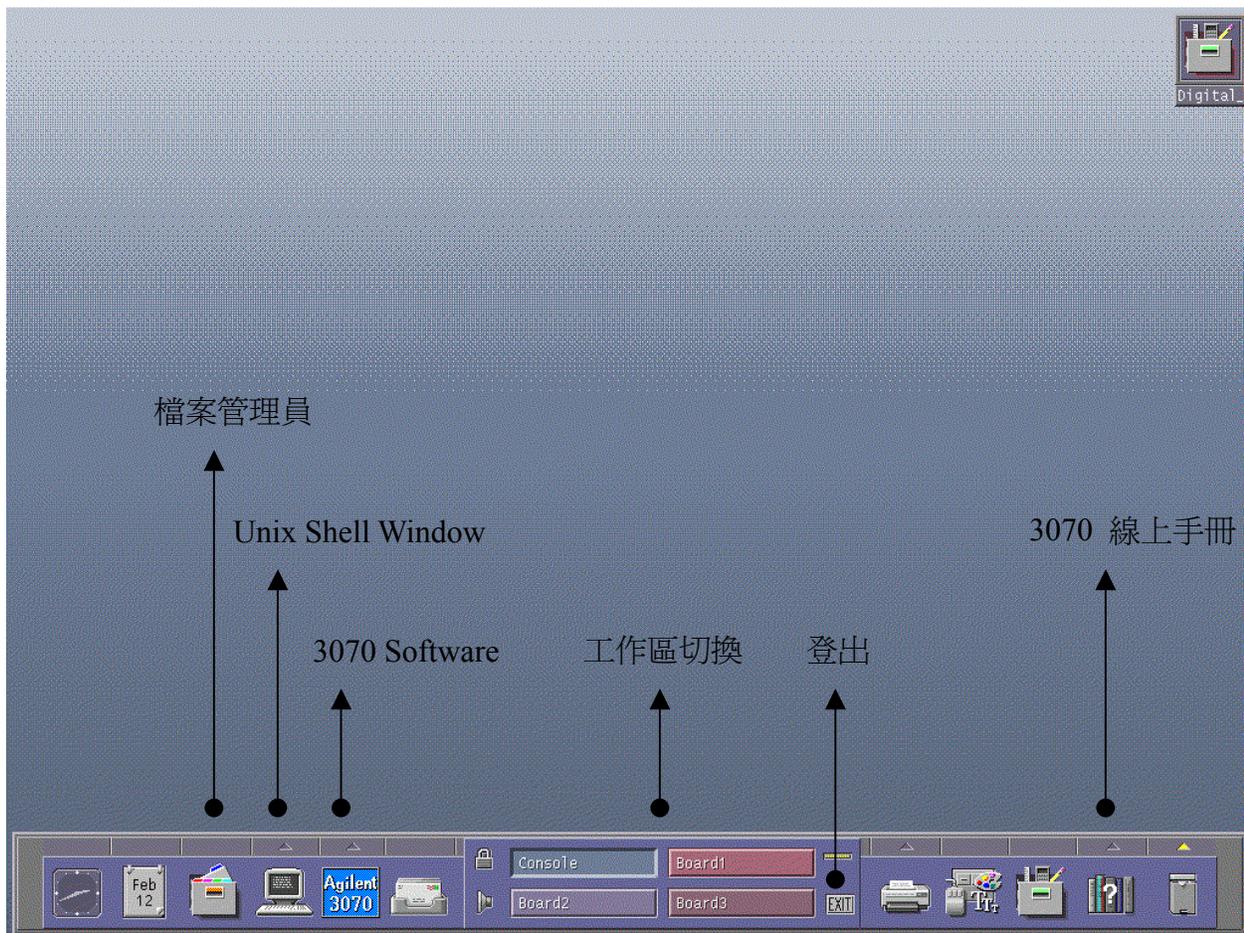


第二節 Agilent 3070 系統軟體簡介

一、 簡介

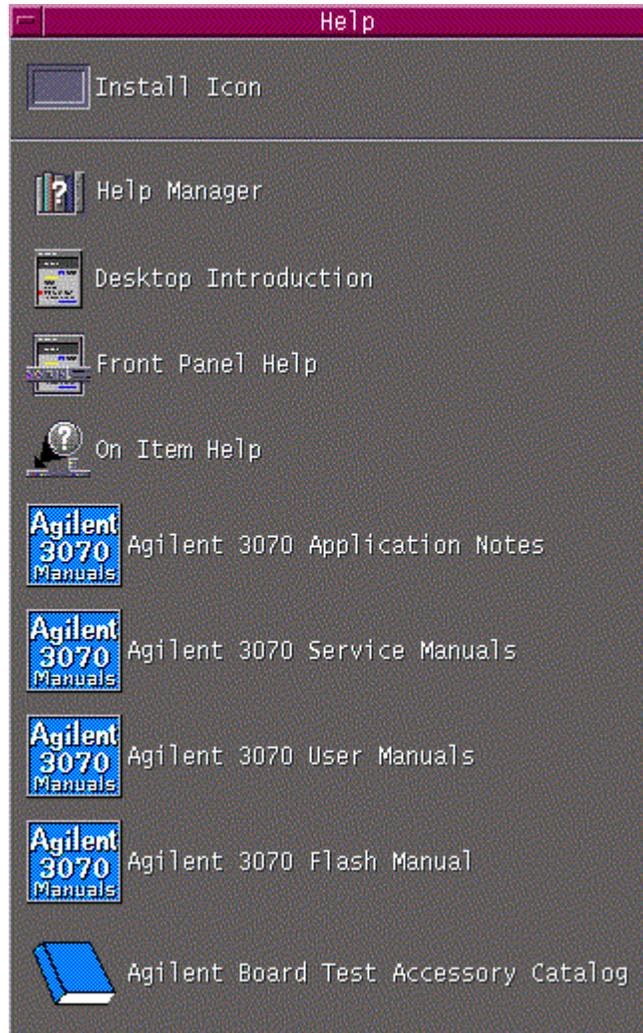
這個章節將介紹 Agilent 3070 操作介面中常使用的工具以及常用的程式，**Agilent Board Consultant**、**Agilent IPG Test Consultant**、**Agilent Fixture Consultant** 及 **Agilent Pushbutton Debug**。

二、 操作介面簡介





三、 線上手冊簡介

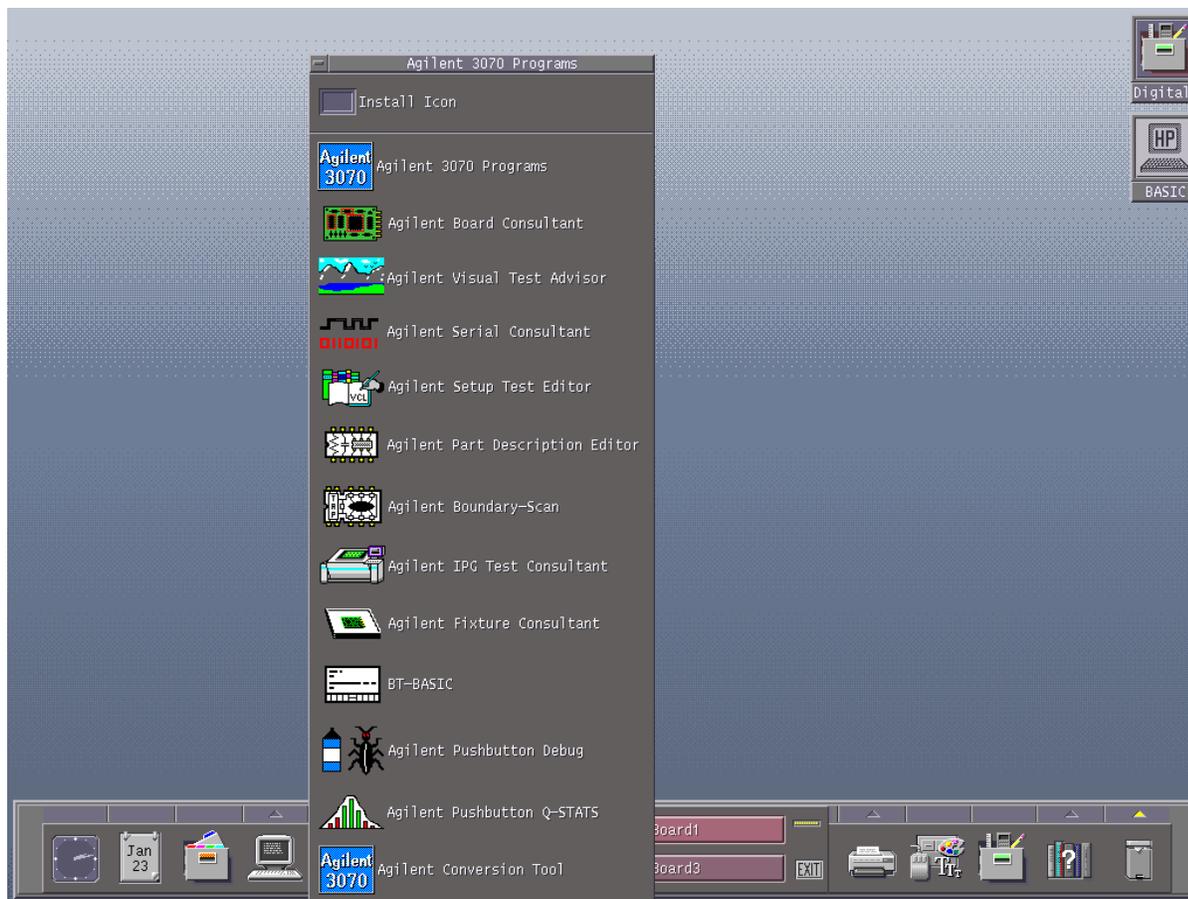


1. **Agilent 3070 Application Notes** : 這是 Agilent 3070 程式應用常見問題。
2. **Agilent 3070 Service Manuals** : 這是介紹 Agilent 3070 硬體的手冊。
3. **Agilent 3070 User Manuals** : 這是介紹 Agilent 3070 軟體的手冊。
4. **Agilent 3070 Flash Manual** : Agilent 3070 Flash Programming(On-Board Programming)手冊。



四、 Board Consultant、IPG Test Consultant、Fixture Consultant 及 Pushbutton

Debug 簡介

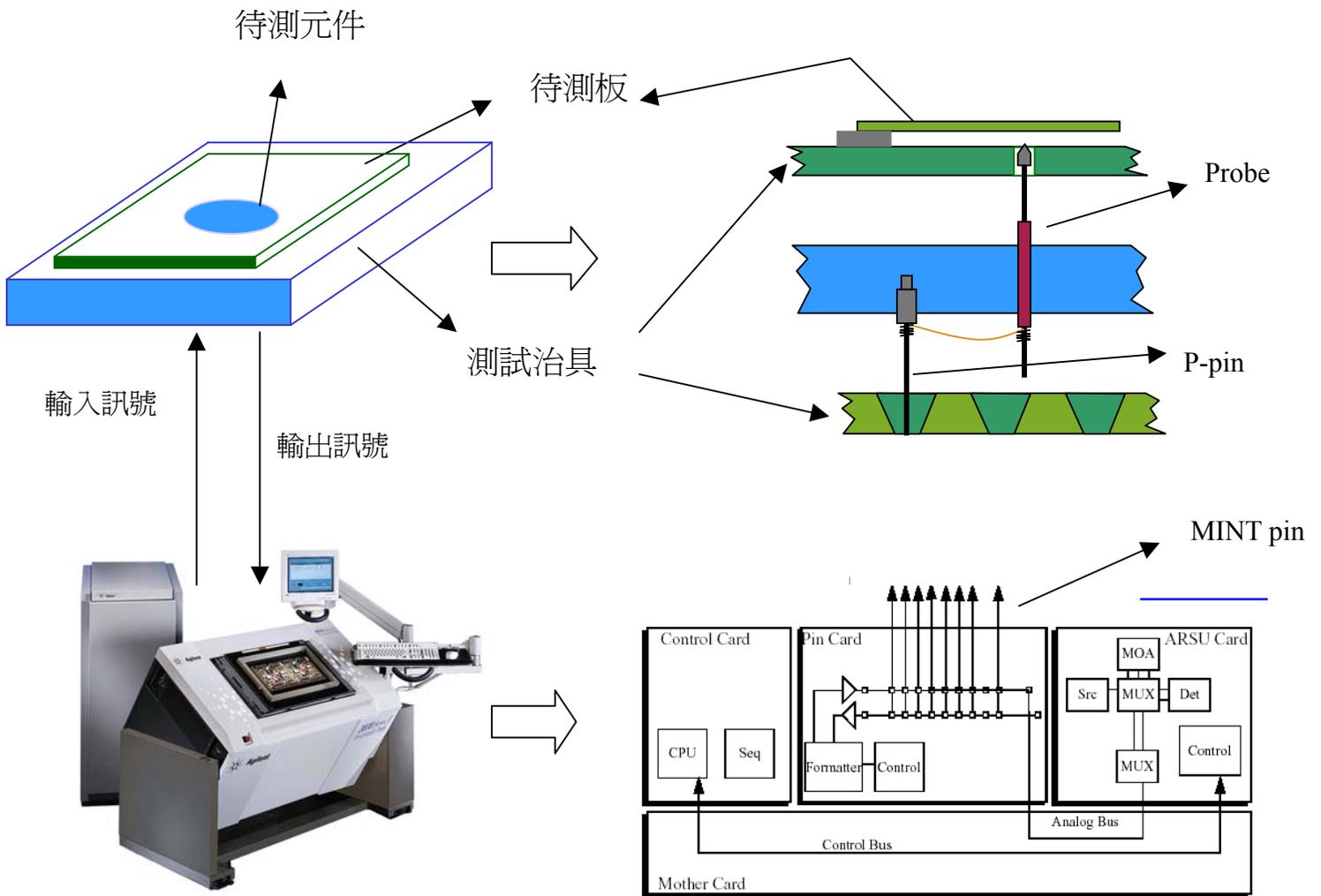


1. **Agilent Board Consultant**：描述待測板所包含的元件以及元件的值、元件名稱、元件測試方法、元件位置、元件與元件間的聯結關係、測試點的位置、待測板的外框、待測板定位孔的位置及待測板定位孔的大小。可以用來尋找待測元件的位置及值。
2. **Agilent Fixture Consultant**：描述待測板治具的資訊。包含待測板與治具間的關係、治具的形式及治具內的繞線的資訊。可以用來尋找探針的位置,繞線的資料及分配 GP relay。
3. **Agilent IPG Test Consultant**：產生所有的測試程式。
4. **Agilent Pushbutton Debug**：這個程式是用來做程式及治具的除錯用的。



第三節 Agilent 3070 測試概念與流程簡介

一、 測試概念



Agilent 3070 系統中的測試資源(resource)如電壓源、電表...等，會經由 pin card 的繼電器 (relay)，MINT pin，治具的 P-pin，繞線和 probe 而連接到待測物上。測試的訊號就是經由這樣的路徑送到待測物上，再回到系統中而完成量測。當要進行下一個元件的量測時，系統會把所有 relay 打開並關閉這個測試所需要的 relay，因此新的路徑又會形成，量測即可進行。



二、 測試流程簡介



Agilent 3070 的測試主程式是”testplan”,在有 testhead 控制權的 BT-BASIC 視窗,開啓”testplan”

按下 F1 或者 Start 就可以測試了.

Agilent 3070 程式的基本測試項目包含未上電測試及上電測試兩大部份，測試順序如上圖所示:

未上電測試:

1. Testing Pin Contact(pins)：測試待測板跟治具間的接觸是否良好(並沒有測試元件的好壞)。這個測試並不一定需要，可視廠內製程來決定。可用 testplan 的“Set_Custom_Options”副程式中“Chek_Point_Mode”旗標來控制，有 Off(不測)、Pretest(每一次測試都先測)、Failure(若有其他測試不良再測)、這三種選擇可用。
2. Testing Preshorts：測試 Jumper、Fuse、Switch 可變電阻等元件。
3. Testing Shorts：測試待測板上應該短路的地方是否短路，應該開路的地方是否開路。
4. Testing Analog Unpowered：測試電容、電阻、二極體、電晶體等元件。
5. Testing Testjet：利用 Testjet 來檢查 IC 的 pin 腳是否有開路。
6. Testing Polarity Check：檢查極性電容是否反向。



7. Testing Connect Check：利用量測 IC I/O pin 的 clamping diode(保護二極體)，來檢查 IC pin 腳是否開路。

上電測試:

8. Setting Up Power Supplies：對待測板上電並檢查這些電源是否正常工作。
9. Testing Digital In-circuit：一般的數位 IC 測試。
10. Testing Digital Functional：數位 IC 的群組(cluster)功能測試，一般少用。
11. Testing Analog Powered and Mixed：量測 IC 的電壓、振盪器的頻率，及數位類比混合型 IC 的測試。



第四節 BT-Basic 使用簡介

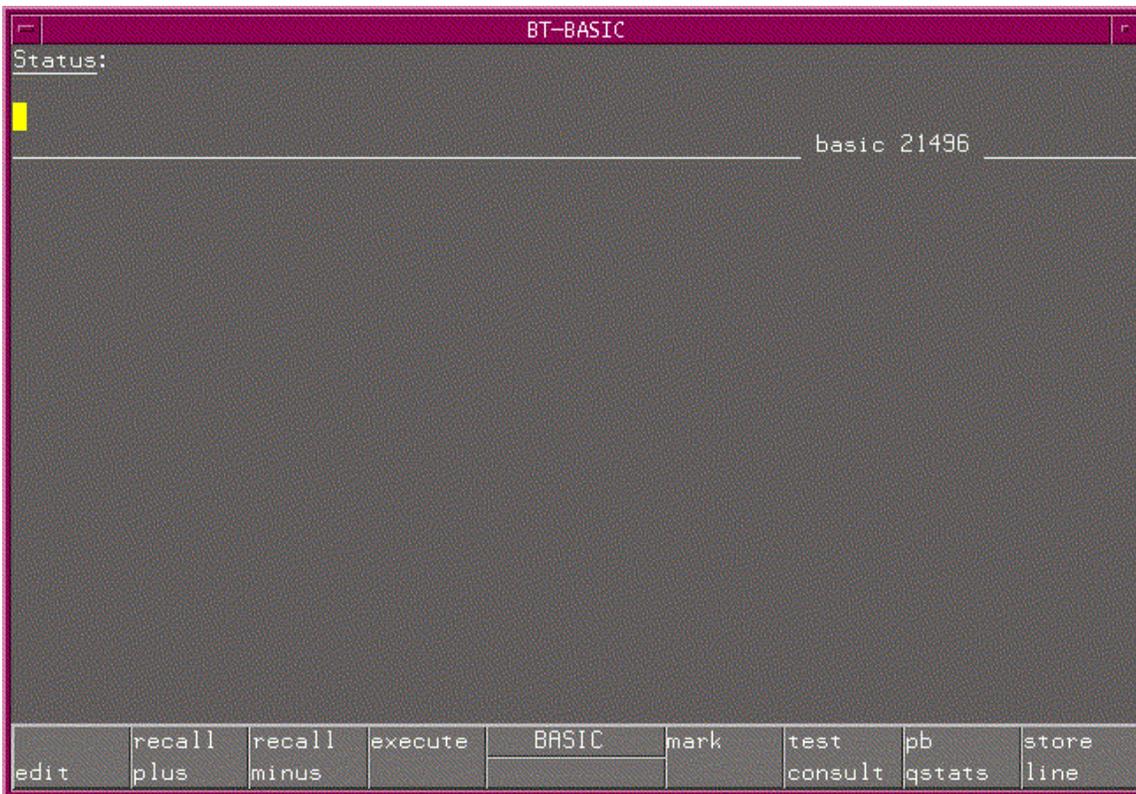
一、 BT-Basic 視窗介紹

BT-Basic 是在 Agilent3070 環境中最基本也是最常用的一個工具，可以用來做程式的編輯，組譯與執行。

狀態列：顯示狀態。

指令列：下達指令時，指令會出現在這一行。

使用模式及程式程序號碼：顯示目前使用的模式及程序號碼。BT-Basic 有各種不同的使用模式如 basic、text、analog...等。在不同的模式下，視窗會用不同的規則檢查輸入的指令與工作區的內容。若要切換視窗的模式，只要輸入想要的模式再按 enter 即可，視窗將會顯示新的模式。



工作區：輸入程式敘述的地方。

編輯與指令互換鍵(F1)：當這個鍵顯示”Edit“時，代表目前使用的是指令列的功能；當這個鍵顯示“Command“時，代表目前使用的是編輯列的功能。



呼叫下一個指令(F2)：BT-Basic 視窗會記錄在這個視窗中曾使用過的指令，可以用這個功能鍵來尋找使用過的前一個指令。

呼叫前一個指令(F3)：這個功能鍵可用來尋找使用過的下一個指令。

執行(F4)：用來執行最後一個被執行過的指令或是目前顯示在指令列的指令。

標記(F5)：可以對某一行或某一區段做標記以便做進一步的修改。按第一下可以標記區間的起始，按第二下可以標記區間的結束，第三下則是取消已標記的區間。

test consult(F6)：用來呼叫 Agilent IPG Test Consultant 程式。

pb qstats(F7)：用來呼叫 Agilent Pushbutton Q-STATS 程式。

儲存目前的指令(F8)：將目前顯示在指令列上的指令或敘述儲存到工作區中。

其他功能鍵：

插入新行(F9)

刪除行(F10)

刪除游標後面的字串(F11)

重新整理 BT-BASIC 視窗畫面(F12)

二、 BT-Basic 常用指令

- ↩ **cat** 列出目前所在路徑下的檔案以及目錄。
- ↩ **change** 將指定的字串以新的字串取代。
- ↩ **changem** 在marked的區間內，將指定的字串以新的字串取代。

範例 1：

```
change 'cards' to 'pin cards'  
changem 'cards' to 'pin cards'
```

- Note**：
1. 將目前的工作區的中的 cards 字串全部改成 pin cards 這個字串。
 2. 將已標記內容中的 cards 字串全部改成 pin cards 這個字串。

- ↩ **copy to** 將檔案複製到新的路徑下，目的檔不存在。
- ↩ **copy over** 將檔案複製到新的路徑下，目的檔已存在。

範例 2：

```
copy 'config' to '/home/user1/class_bd/config'  
copy 'config' over '/home/user1/class_bd/config'
```



- Note :**
1. 執行 copy to 的指令，將目前目錄的 config 檔案拷貝到/home/user1/class_bd 中並以 config 為檔名儲存。
 2. 若/home/user1/class_bd/config 已存在，則需使用 copy over 以新檔蓋過舊檔。



↵ **creat dir** 產生新的目錄。

範例：

```
creat dir 'class_bd'  
creat dir '/home/user1/class_bd'
```

Note：1. 在目前的工作目錄下產生一個叫做 class_bd 的子目錄。
2. 在/home/user1 的目錄下產生一個叫做 class_bd 的子目錄。

↵ **delete** 將用<F5>標記出來的區間內的資料刪除。

↵ **duplicate** 複製用<F5>標記出來的區間。

↵ **edit** 移動游標到指定的行數。若未指定，則為第一行。

範例：

```
edit 27
```

Note：將游標移動到第 27 行。

↵ **fetch** 到指定的行數並將這一行複製到指令列。

↵ **find** 找出在檔案中的第一個指定字串。

↵ **findn** 以目前游標位置為基準,找下一個指定的字串。

範例：

```
find 'cards'  
findn 'cards'
```

Note：1. 找尋目前的檔案中的第一個 cards 字串。
2. 找尋目前的檔案中的下一個 cards 字串。

↵ **get** 載入檔案。

範例：

```
get 'config'  
get 'board';win
```

Note：1. 將 config 檔案開啓在 BT-Basic 視窗中。
2. 將 board 檔案用一個新的 BT-Basic 視窗開啓。

↵ **list** 在檔案前面加上行數號碼。

↵ **load** 載入檔案，用法同”get”。



↵ **merge** 將指定的檔案合併到現在的檔案內。

範例：

```
merge 'file_test',1,10
```

Note： 將 file_test 檔案中的第 1 行到第 10 行拷貝出來並插入目前使用中檔案的游標所在位置。

↵ **mov** 將用<F5>標記的區間移動到游標所在位置。

↵ **msi** 更改目前的工作路徑。

↵ **msi\$** 列出目前的工作路徑。

範例：

```
msi '/home/user1'
```

```
msi$
```

```
/home/user1
```

Note： 1. 將目前的工作目錄指到 /home/user1 這個目錄。
2. 顯示目前所在的工作目錄。
3. 執行 msi\$的結果。

↵ **number** 列出目前游標所在的行數。

↵ **scratch** 將BT-Basic視窗中的所有內容全部清除。

↵ **save** 將目前的資料儲存。

↵ **re-save** 再次儲存檔案（會將原來的檔案蓋過）。

範例：

```
save "/home/user1/class_bd/board"
```

```
re-save
```

↵ **unlink** 刪除檔案。

範例：

```
unlink "testplan"
```

Note： 刪除目前路徑下 testplan 這個檔案。

↵ **testhead is 1** 取得testhead 控制權。

↵ **testhead is *** 釋放testhead 控制權。

↵ **exit** 結束關閉BT-BASIC 視窗。



第三章 基本操作

第一節 如何載入治具及測試程式

1. 將治具放在 Agilent 3070 Testhead 上。
2. 在擁有 testhead 控制權的 BT-BASIC 視窗中輸入 **fixture lock** 並按下鍵盤上的 **Enter** 鍵。
3. 在 BT-BASIC 視窗中輸入 **msi "program_path"** 再按下鍵盤上的 **Enter** 鍵，將路徑切換到待測板程式路徑。其中 **program_path** 為程式所在位置，如 `/home/user1/class_bd`。
4. 輸入 **get 'testplan'** 按下鍵盤上的 **Enter** 鍵。
5. 輸入 **run**。
6. 即可開始測試。

如果要更換治具及程式呢？

1. 在控制 testhead 的 BT-BASIC 視窗輸入 **fixture unlock** 並按下鍵盤上的 **Enter** 鍵。
2. 取下治具,再換上待測機種的治具。
3. 在原來的 BT-BASIC 視窗中輸入 **msi "program_path"** 再按下鍵盤上的 **Enter** 鍵，將目錄移到新的程式位置。
4. 輸入 **fixture lock** 按下鍵盤上的 **Enter** 鍵。
5. 輸入 **get 'testplan'** 按下鍵盤上的 **Enter** 鍵。
6. 輸入 **run**。
7. 即可開始測試。



第二節 如何解讀 error ticket

以下將針對 3070 各種不良報表做簡單的解釋與說明：

1. shorts

```
-----
Open #1 Thresh 9, Delay 50us      Ohms
From: J201-1                      22072  Open
To:   J201-2                      22073
Message is:  None.
```

Open 代表開路不良，
節點 J201-1 與 J201-2 之間應該有短路(小於臨界阻抗 9ohm) ，待測板上卻量測為開路(>9ohm),
22072 代表節點 J201-1 所連接的 MINT pin(BRC) 。

```
-----
Short #1, Thresh 1000, Delay 50us  Ohms
From: CR201-C                      22163   3
To:   VR201-C                      22162   3
Total of 2 nodes, Message is:  None.
```

Short 代表短路不良
節點 CR201-C 與 VR201-C 之間應該是開路(臨界阻抗 1000ohm) ，待測板上卻量測為短路，量
測值為 3ohm ，小於臨界阻抗。

2. analog in-circuit

```
-----
r205 HAS FAILED
Measured:  1.9413M
Nominal:   1.0000k
High Limit: 1.0836k
Low Limit:  962.20
Resistance in OHMS
```

電阻不良，R205 應該是 1kohm 的電阻，上限 1.0836kohm，下限 962.2ohm，卻量測到
1.9413Mohm。



L201 HAS FAILED

Measured: -17.153
Nominal: 10.000m
High Limit: 10.897m
Low Limit: 9.8110m
Inductance in HENRYs

電感 L201 不良。

cb313 HAS FAILED

Measured: -2839.8p
Nominal: 66.660u
High Limit: 72.086u
Low Limit: 54.061u
Capacitance in FARADS

DEVICES IN PARALLEL

cb314 22.0u
cb315 22.0u
cb301 100n
cb302 100n

電容 CB313 不良，並聯元件有 CB314、CB315、CB301、CB302。

One classy board

Mon Apr 21 10:02:38 2003

ac307

Compensation capacitance has not been
learned

c202

Compensation capacitance has not been
learned



某些小電容測試會先 learn 系統跟治具的寄生電容以便在測試時做補償計算,而這些量測值會被記錄在”.o”檔中,如果這些值被清掉了(例如:該測試被重新 compile),則測試就會出現以上訊息.要排除此錯誤,只要重新 get”testplan” 再 run 即可.

d1 HAS FAILED

Measured: 1.4187

High Limit: 828.00m

Low Limit: 213.00m

Forward VOLTAGE drop

Status: 2cH

Moa Voltage/Current Compliance

Detector Over Range

Integrator Over Voltage

二極體 D1 不良,導通電壓上限 828mV、下限 213mV，量測值為 1.4187V。

q201 HAS FAILED

Subtest: B-E

Measured: 1.4048

High Limit: 806.00m

Low Limit: 603.00m

Forward VOLTAGE drop

Status: 2cH

Moa Voltage/Current Compliance

Detector Over Range

Integrator Over Voltage

1:q201 HAS FAILED

Subtest: B-C

Measured: 1.4048

High Limit: 806.00m

Low Limit: 603.00m

Forward VOLTAGE drop

Status: 2cH



Moa Voltage/Current Compliance
Detector Over Range
Integrator Over Voltage

電晶體 Q201 不良，B-E 極 B-C 極導通電壓皆量測不良。

3. testjet

Open #1 Device u1
Pin 1 Node DVOCLKOUT
Measured 2.0 (BRC 21423)
Open #2 Device u1
Pin 10 Node DRV2A
Through c12
Measured 3.6 (BRC 214104)

元件 U10 第 1 腳及第 10 腳不良，第 1 腳之節點為 DVOCLKOUT 量測值為 2fF。
若有使用 drive through 則如第 10 腳之訊息，由節點 DRV2A 送入訊號經過 C12 來量測,量測值為 3.6fF。

4. power supply

Power Supply Number
2
In Current Limit

Check for backwards
IC's or Capacitors.

電源過電流錯誤訊息。
power supply number 2 已送出所指定之最大電流，可是仍未達到所指定之電壓。



5. digital test

u301 HAS FAILED

vector = 32

user vector = 10

source line = 271

source file = digital/u301

Status: 15H

Pass/Fail error on following pins:

BRRCC	NODE	PIN
21444	RIPPLE	U301.15

- ← 執行時第 32 支 vector
- ← 程式描述中第 10 支 vector
- ← 程式中的第 271 行

← 此 pin 接收到與預計要接收之訊號不同

u302 HAS FAILED

vector near 52

user vector near 28

source line near 331

source file = digital/u302

Status: 1041H

Software timeout occurred and forced
the sequencer to stop

Clocks were on when stopped

← 一直沒有滿足 homingloop 之條件,
使程式無法跳出

元件 U301,U302 不良

Note1：因為程式中有 homingloop 或 repeat loop，使得同一支 vector 在程式中和實際執行時會有不同的序號。

Note2：數位測試較為複雜,error ticket 所列之 pin 腳不一定就是不良，而是表示這支 pin 腳接收到之訊號與所期望之訊號不同。



6. power analog

u17 HAS FAILED

Subtest: output

Measured: 17.168m

← 量測到之電壓

High Limit: 5.5000

← 電壓上限

Low Limit: 4.5000

← 電壓下限

Direct Current VOLTS

y1 HAS FAILED

Subtest: out

Measured: 10.357M

← 量測到之頻率

High Limit: 15.000M

← 頻率上限

Low Limit: 13.500M

← 頻率下限

Frequency in HERTZ

y2 HAS FAILED

Subtest: out

Measured: 2.0000e+100

High Limit: 73.500M

Low Limit: 66.500M

Frequency in HERTZ

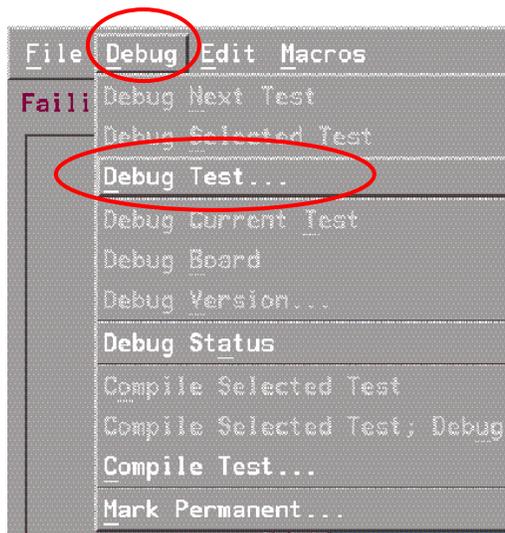
Status: 2000H

Timeout waiting for measurement complete ← 表示完全沒有量到頻率

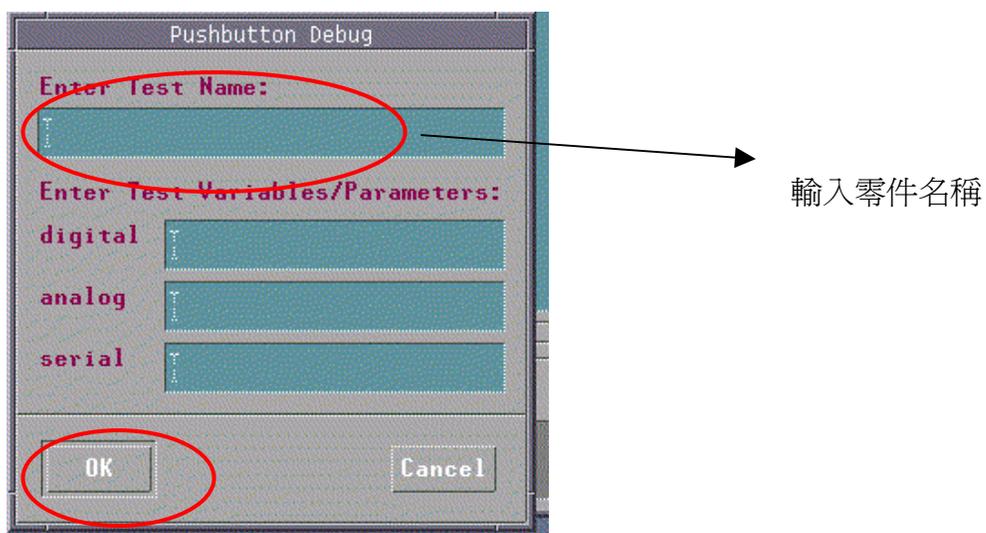


第三節 如何 debug analog 元件

1. 在testplan的視窗，執行load board ldebug board 開啓Push Button Debug視窗。
2. 按下faon。
3. 在BT-BASIC中，輸入 unpowered 按 。
4. 在Push Button Debug 視窗,點選Debug→Debug Test→。



5. 輸入要debug 的零件，如：**analog/c1**，按ok，進入測試程式。





6. 測式程式視窗

File Debug Edit Macros Mode **Execute** Display Help

File:*bd_shopflow/analog/c201 Analog

```
! IPG: rev B275 Tue Aug 05 17:12:32 1997
! Common Lead Resistance 500m, Common Lead Inductance 1.00u
! Fixture: EXPRESS
disconnect all
connect s to "C201-1"
connect i to "C201-2"
capacitor 10n, 10.9, 10.2, fr1024, re4, wb, ar100m, nocomp
! "c201" test.
! DUT: nominal 10n, plus tol 10.0 %, minus tol 10.0 %
! DUT: high 11n, low 9n
! TEST: high limit 11.087n, low limit 8.981n
! Tolerance Multiplier 5.00
! Remote Sensing is Allowed
```

Relay 控制

測式命令

Messages: Device Level Debug Break Command

Complete.

capacitor 10n ,10.9,10.2,fr1024,re4,wb,ar100m,nocomp

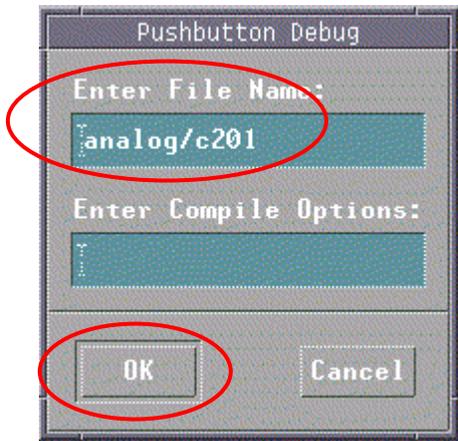
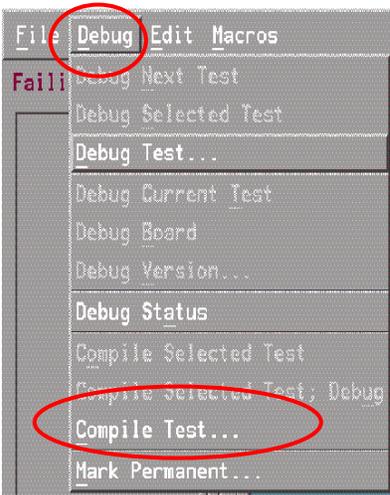
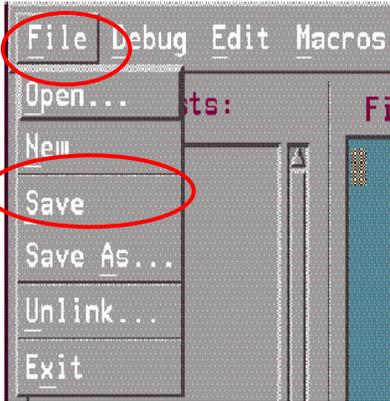
期望值 上限 下限 Options

量測值應在 $10n + 10n * 10.9\%$ 以及 $10n - 10n * 10.2\%$ 之間，其中 $10n$ 為BOM中的值，而 10.9 及 10.2 即為上下限的tolerance。

7. 執行Execute → Execute to Fail，可以看到執行結果。
8. debug過程中，如果有修改測試程式，可以按Compile and Go，看執行結果，這個功能鍵會對程式做編譯加上執行。

Note: Compile and Go 只針對debug 視窗中的程式做編譯及執行，並沒有更新或改變原始檔和.o檔。

9. debug 完成後，要re-save (FILE→Save)，compile (Debug→Compile Test)不論任何變動或調整，請務必記得一定要re-save 及compile，否則，還是會執行到舊的執行檔。





Analog元件測試調整

1. 將s,i bus 互換，若有使用a,b bus則也要一起互換。
2. 調整上限及下限。
3. 嚐試拿掉所有或部份的guarding。
4. 加ed 或wait time。

範例：

resistor 100k, 12,9.5,re5, aml20m,ed ,wa15m

Note：使用ed會使測試時間增長至17.5ms。

5. 如果如何調整都不行，那麼可以在testplan中先將該零件測試程式項目前加“!”，暫時忽略。

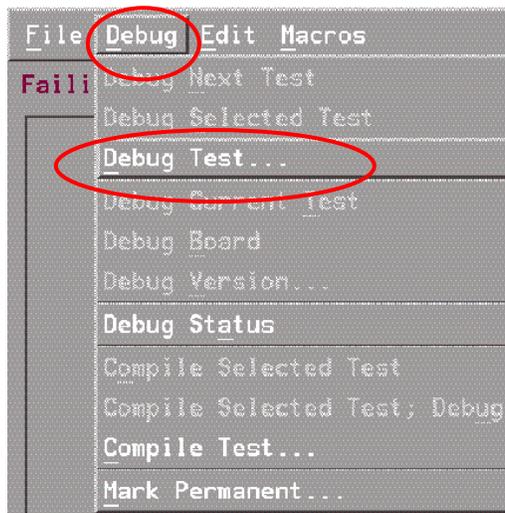
例如: testplan中

```
test" analog/c1"  
test" analog/c2"  
!test"analog/r1" →如果r1 測試前有“!”，執行testplan時，r1就不會被測試  
test"analog/r2"  
test"analog/r3"
```

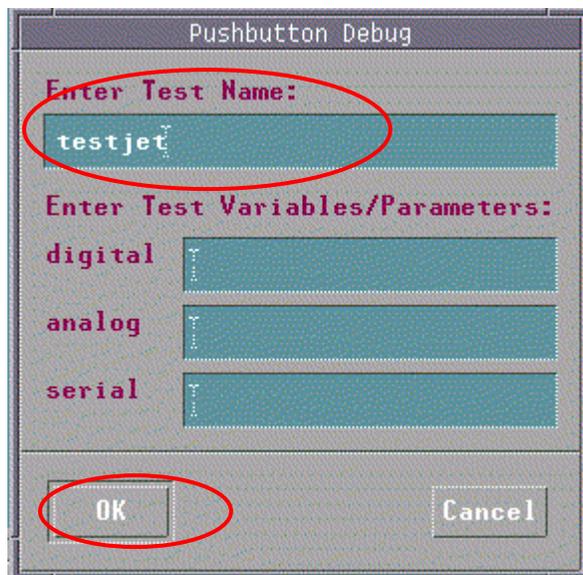


第四節 如何 debug testjet

1. 在testplan的視窗，執行load board ldebug board 開啓Push Button Debug 視窗。
2. 按下faon。
3. 在Push Button Debug 視窗，點選Debug→Debug Test→

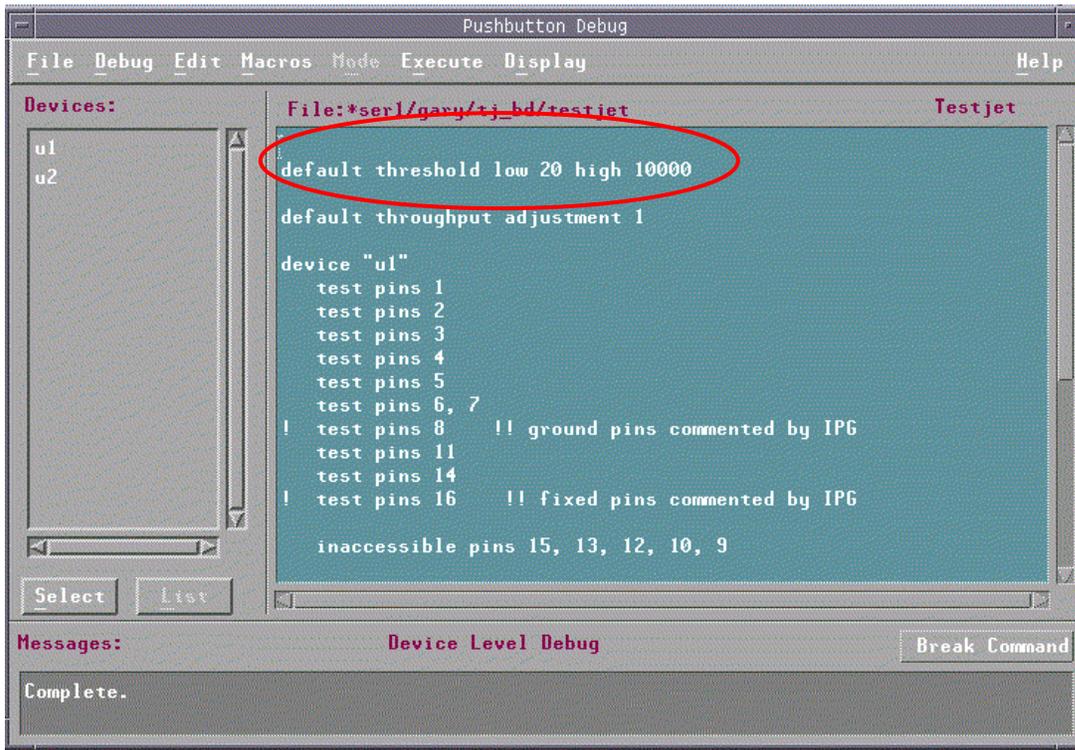


4. 輸入檔名：testjet





5. 進入testjet測試程式：



紅色框的是default threshold，這個threshold 適用於testjet 檔案中每一個零件，如果只想針對某些pin修改 threshold，只要將欲修改的上下限加在該pin的後面即可。

例如：

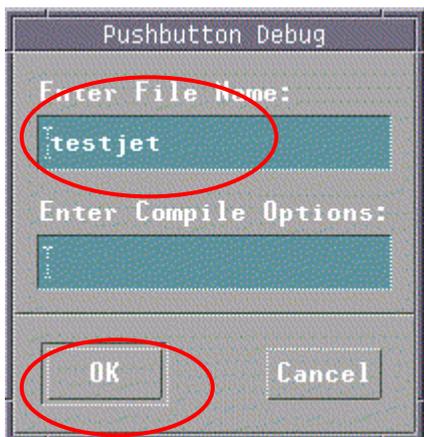
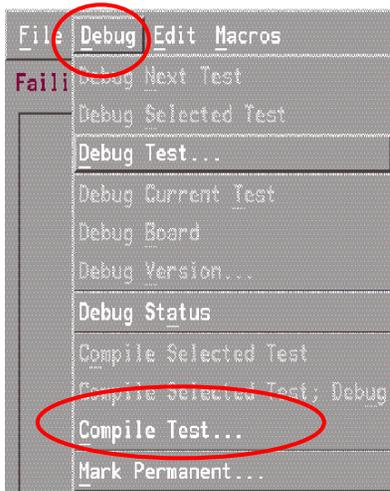
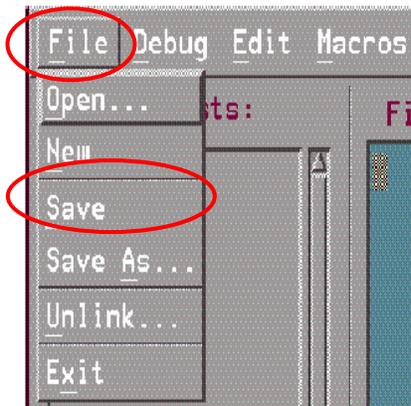
```

default threshold low 20 high 10000
default throughput adjustment 1

device "u1"
  test pins 1 ;threshold low 90 high 433
  test pins 2 ;threshold low 85 high 450
  test pins 3 ;threshold low 80 high 412
  .
  .
end device
    
```



6. debug 完成後,要re-save (FILE→Save) , compile (Debug→Compile Test)不論任何變動或調整,請務必記得一定要re-save 及compile,否則,還是會執行到舊的執行檔。





根據error ticket 調整誤判的pin

根據error ticket 上所列印的量測值，修改誤判pin 的 下限tolerance，使不良的值落在上限跟下限之間，就不會刷出來了。

例如:error ticket :

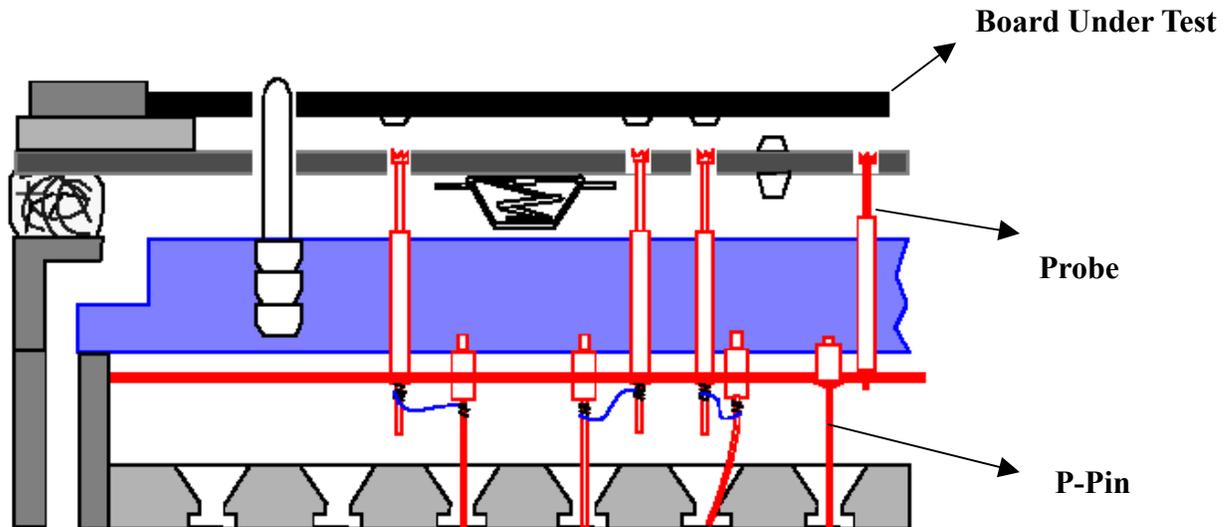
```
-----  
TestJet Report for "testjet".  
Wed Feb 14 10:58:44 2001  
These are the board defaults  
-----  
Open #1 Device u1  
  Pin 1 Node U1-1  
  Measured 80 (BRC 21423)  
Open #2 Device u1  
  Pin 2 Node U1-2  
  Measured 70 (BRC 21404)  
Open #3 Device u1  
  Pin 3 Node U1-3  
  Measured 85  
-----
```

Testjet 程式:

```
device "u1"  
  test pins 1 ;threshold low 70 high 433  
  test pins 2 ;threshold low 60 high 450  
  test pins 3 ;threshold low 76 high 412  
  .  
  .  
  .  
end device
```



第五節 治具的維護與保養



一、 如何利用 **fixture consultant** 找到某一節點(node)之探針(probe)並更換

1. 先找到節點的名字(node name) ，例如要找 R314 的第二支腳，可以用 bt-basic 打開 board file ，用 find “r314.2”來找到這個 node name 。

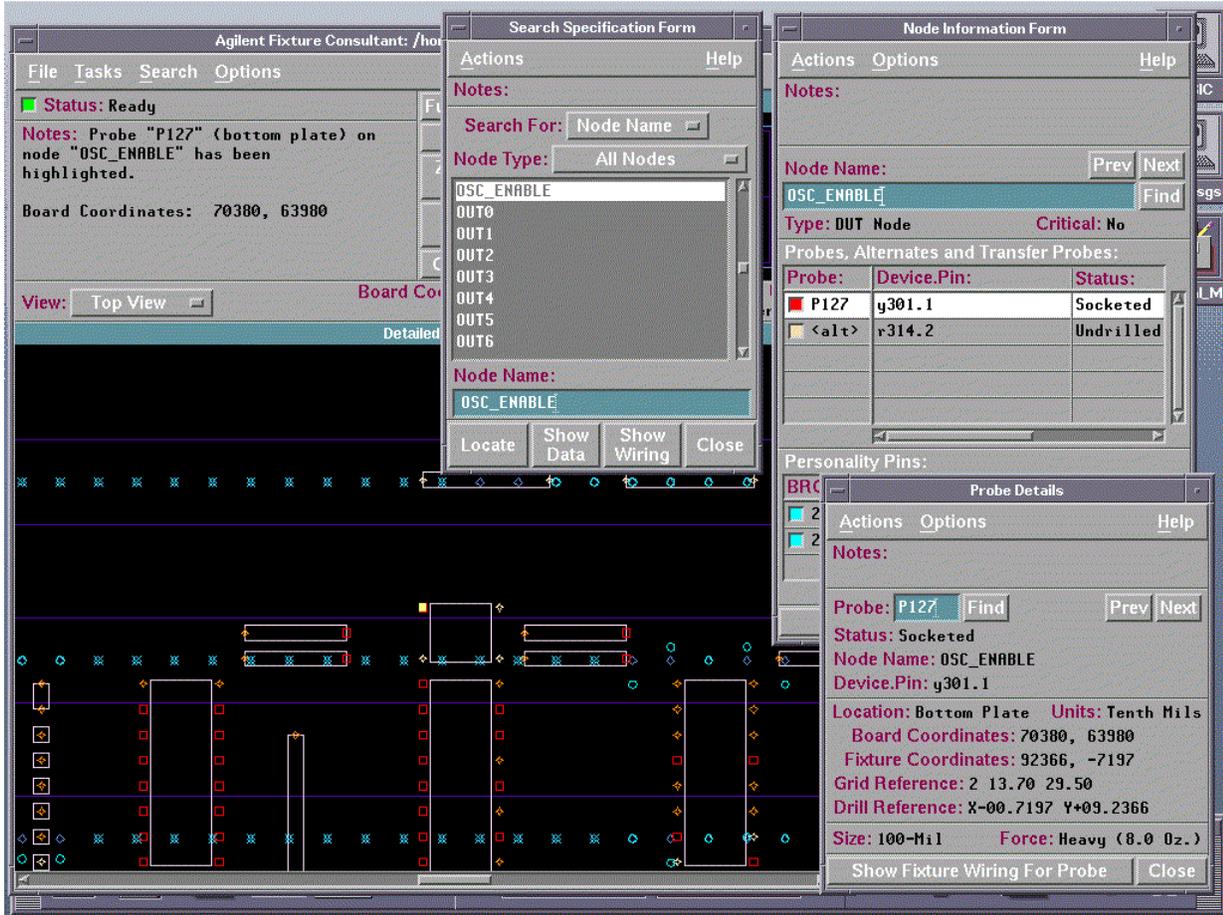
```

board
-----
Status:
find "r314.2"
text 8979
NC1
U302.4;
NC2
U302.6;
NC3
U302.8;
NC4
U302.12;
NC5
U302.14;
NC6
U302.16;
SC_ENABLE
R314.2
Y301.1;
OUT0
U303.6
U306.2;
OUT1
U303.2
command recall recall execute board mark test pb store
plus minus minus consult qstats line
    
```

2. 打開 **fixture consultant**, **Search/For Node...** ，輸入 node name 然後按 **Show Data** ，會打開一個新的視窗 **Node Information Form** ，用右鍵點選 probe ，則 probe 的位置會顯示在視窗中。



Note: fixture consultant, Options/Features Displayed... 可以用來關閉一些不相關的圖形以便尋找，另外也可以用 **Top View/Bottom View** 來切換視角。



二、 如何換針？

注意針的大小以及形狀。

1. 常用的針有50mil, 75 mil, 100mil。
2. 平的測試點用尖針或有剖面的針。
3. 突起的測試點用鈍針或爪針。
4. DIP零件的腳當測試點則用爪針。



三、 常用的指令：

```
find pins
find testjet probes
verify all nodes
verify "node_name"
    輔助按鍵
    F1    Start
    F2    Skip Node
    F3    Re-try Node
    F6    Show all dig gnds / show one dig gnd
    F7    Auto Start / manual Start
    F8    End the Verify process
```

```
verify testjet probes "device_name"
verify all mux cards
```

請參考實驗三 Find Pins 及Verify Node練習。

四、 日常保養

這些可以列入ATE日常保養項目中。

1. Testjet 的sensor plate與IC的接觸面，如果髒了要擦拭，否則會影響量到的值。
2. 有時因製程關係probe 上容易堆積flux，這樣可能會造成測試誤判，應常常以乾淨而不沾任何清潔溶劑的刷子清理，**切記!不可以用鋼刷**，否則清理過程中如果刷毛脫落，常發生不明原因短路。
3. 測試次數多了，治具內容易有錫珠、錫渣、灰塵之類的髒東西，保養時可用氣槍(airgun)，將髒東西沖掉。

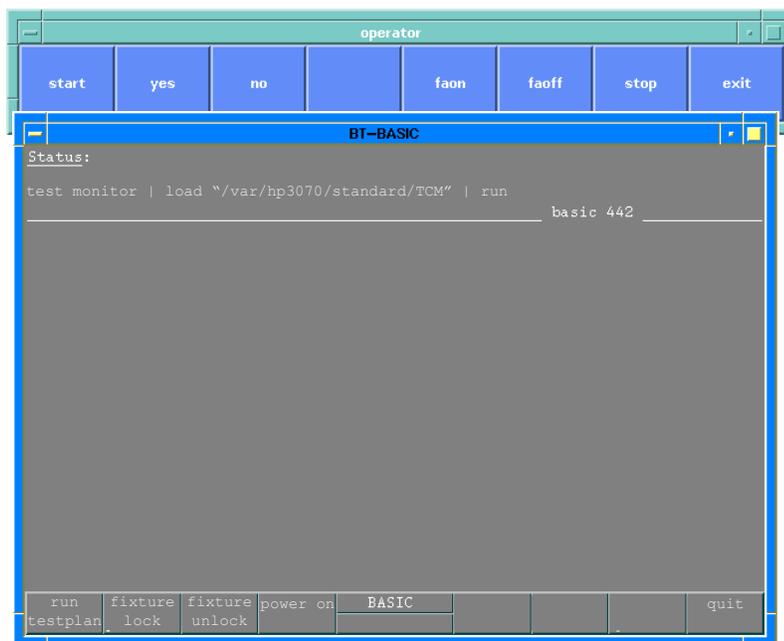


第六節 如何使用 Operator Mode

一、 登入與操作

1. 在 unix 登入時，使用身分 **operator**。
2. 登入後會出現一個 **bt-basic** 視窗可使用功能鍵來執行以下功能。
 - F1 --- run testplan
 - F2 --- fixture lock
 - F3 --- fixture unlock
 - F4 --- testhead power on
 - F8 --- quit

Operator Mode



3. 系統將自動 power on testhead。
4. 按下 F1，則系統將自動鎖下治具(fixture lock)，切換路徑到治具對應程式所在位置下，並執行 testplan。
5. 按 F8 則可退出。



二、 autofile 的修改

1. 每一個治具都會有一個獨特的 ID number，這個 number 可以在 fixture consultant 裏找到 (Tasks/View/Edit Fixture Options/Autofile)，並且有特殊的繞線使得治具被放到 testhead 上，機台就可以偵測到這個 ID number。在程式發展過程中，系統會自動產生一個檔案在 /hp3070/autofile 的路徑下，並以這個 ID number 命名，內容包含 board name 及測試程式所在位置。例如有一個治具的 ID number 是 3094，board name 是 class_bd，程式放在 /home/user1/class_bd，則/hp3070/autofile/3094 的內容會如以下：

```
BT-BASIC
Status: 2 lines in workspace.
get "/hp3070/autofile/3094"
class_bd
/home/user1/class_bd
```

edit	recall plus	recall minus	execute	BASIC IC	mark	test consult	pb qstats	store line
------	----------------	-----------------	---------	-------------	------	-----------------	--------------	---------------

若這個設定有問題則 operator mode 就不能正常工作。因此，若測試程式被搬移到別的位置則要來修改這檔案。



第七節 常見之錯誤訊息

一、 BT-Basic

1. F9 The file or directory was not found

通常是出現在 get 或 msi 之後，意指找不到所指定的檔案或子目錄，應查明路徑和檔名之後再試一次。

2. R214 Testhead required

所執行的指令需要 testhead，例如 unpowered,faon...這些指令需要在擁有 testhead 控制權的視窗才能正常執行。

3. R222 Must execute an "unpowered" statement

這個測試需要先執行 unpowered 指令，只要在 bt-basic 中先執行 unpowered 即可排除。

4. R233 Board object could not be loaded

這個訊息代表 board.o 不存在或系統無法辨識，通常是因為 board.o 是由較新版本軟體產生的，使得 load board 指令無法完成(load board 會將 board.o,shorts.o,wirelist.o 先存放到系統記憶體中以便讀取)。只要將 board 重新再 compile 一次即可。

5. R252 Cannot "save/store" to existing file; use "re-save/re-store"

save 指令所指定之檔案已存在，需使用 re-save 再執行一次。

6. R255 "get/load" not done since working directory changed

此訊息發生在 re-save 時，是因為當 get 檔案之後又使用 msi 切換到另一個路徑，因此無法執行 re-save。只要在 re-save 之後指定絕對路徑及檔名即可回存。

7. R533 Cannot access debug object

digital debug 會需要 .d 檔存在否則無法啟動。要產生 .d 只要在 compile 後加 debug 參數，例如 compile“digital/u301”;debug(這個指令會同時產生 .d 及 .o)。

8. R759 Configuration object file missing or unreadable.

Config.o 不存在或無法辨識，重新 compile 即可。

9. Could not open board.o. (ERROR DGC65)

測試程式的 compile 需要在 board directory 下執行，例如 compile“analog/r17”。因為 compile 時需要和 board.o 做比對，若 compile 時的路徑不是在 board directory 下，系統找不到 board.o 則會有這個訊息。



10. The workspace is edited; press softkey f4 to really exit.

這個訊息出現在要退出 bt-basic 視窗時。這是因為視窗內容已經改變而尚未回存。若要放棄修改內容並結束請按 F4，若要回存則按其他鍵結束 exit 指令後再執行 re-save。

11. No autofile for autofile code: 4095

這個訊息發生在 operator mode.當系統偵測到的 fixture ID 在/hp3070/autofile 下不存在時，則有這個訊息。如以上的例子則是系統偵測到 fixture ID 為 4095 可是 /hp3070/autofile/4095 卻不存在。

二、 Unix

1. tar: cannot open /dev/rmt/0m

磁帶機尚未準備好，稍等一下再重試。

2. mkdir: cannot access .: Permission denied

下 mkdir 指令的用戶沒有權限在指定路徑下建立新目錄。

3. cp: cannot create ./board: Permission denied

下 cp 指令的用戶沒有權限在指定路徑下建立新檔案。

4. sh: home: not found.

這個訊息發生在 cd home 之後，系統在目前路徑下找不到 home 這個目錄。

5. cp:temp/: No such file or directory

這個訊息發生在 cp testorder temp/testorder 之後，系統在目前路徑下找不到 temp 這個目錄。



第四章 實驗

實驗一 Software Configure 設定

Config 檔記錄系統卡片的配置，可分為硬體 config 和軟體 config，詳情請參考以下 Note10，軟體的 config 檔位於每個程式的目錄中。這個實驗示範如何建立一個 software config 檔。

1. ____ 登入工作站後，開啓一個 BT-BASIC 視窗。
2. ____ 在指令列輸入 `msi$` 指令，再按”`↓Enter`”按鍵來得知目前的工作路徑（如同 Unix 下的 `pwd` 指令）
3. ____ 在指令列輸入 `cat` 指令，再按”`↓Enter`”按鍵來得知目前的工作路徑下有的檔案及目錄（若加上 “`short`” 這個選項可以用較短的資訊來表示目前的工作路徑下有的檔案及目錄。如 “`cat ; short`”）
4. ____ 建立一個子目錄，在指令列輸入 `creat dir “class_bd”` 指令，再按”`↓Enter`”按鍵來建立一個叫做 `class_bd` 的子目錄。
5. ____ 在指令列輸入 `cat ;short` 指令，再按”`↓Enter`”按鍵來得知目前的工作路徑下有的檔案及目錄（您會發現您已經有多一個叫 `class_bd` 的目錄了）
6. ____ 在指令列輸入 `get “/var/hp3070/diagnostics/th1/config”` 指令，再按”`↓Enter`”按鍵。來將 `/var/hp3070/diagnostics/th1/config` 這個檔案開啓。（用 `load` 這個指令也可以達到同一個目的）

Note：這個 config 檔案是機器的硬體 config 檔（H/W config）

7. ____ 在指令列輸入 `msi “/home/user1/class_bd”` 指令，再按”`↓Enter`”按鍵。來讓目前的工作目錄進入 `/home/user1/class_bd` 下。
8. ____ 在指令列輸入 `configuration` 指令，再按”`↓Enter`”按鍵。（這時 `bt-basic` 的使用模式會從 `basic` 變成 `configuration`）
9. ____ 在工作區鍵入以下的敘述：

```
target hp3070 standard    ! Target HP3070-6.25 Mhz system
! target hp3070 advanced  ! Target HP3070-12.5 Mhz system
```



! target hp3070 accuracy ! Target HP3070-20 Mhz system
enable software revision b
enable express fixturing
module 3
 cards 1 asru
 cards 2 to 5 hybrid standard
 cards 6 control
 cards 7 to 11 hybrid standard
 supplies hp6624 1 to 4 asru channels 1 to 4
 ports ext1 , ext2
end module

```

BT-BASIC
Status:
_____ configuration 9664 _____
target hp3073 standard
enable software revision b
enable express fixturing
module 3
  cards 1 asru
  cards 2 to 5 hybrid standard
  cards 6 control
  cards 7 to 11 hybrid standard
  supplies hp6624 1 to 4 asru channels 1 to 4
  ports ext1, ext2
end module

```

command	recall	recall	execute	BASIC	mark	test	pb	store
	plus	minus				consult	qstats	line

10. ____ 在指令列輸入 save ‘ config ’ 指令，再按“↵Enter”按鍵。(系統會以 config 這個名稱來儲存這個檔案)
11. ____ 在指令列輸入 compile ‘ config ’ 指令，再按“↵Enter”按鍵。(系統會組譯這個檔案並產生 config.o 這個檔案)
12. ____ 在確定沒有 ERROR 後，確定系統已經組譯成功。



Note :

1. target hp3070 standard : 在 Agilent 3070 系統中，有不同的測試速度可供選擇。
hp3070 standard 可提供的測試速度為 6Mhz ; hp3070 advanced 可提供的測試速度為 12Mhz ;
hp3070 accuracy 可提供的測試速度為 20Mhz ，這要視系統的規格來調整。
2. enable software revision b : 啓動 3070 B 版的軟體。
3. enable express fixturing : 啓動軟體來產生短繞線的治具。
4. cards 1 asru : 告知系統您目前的 testhead 使用的 asru 卡是 A 或 B 版本。如果您使用的是最新的 C 版本，那麼您的敘述要改為 **cards 1 asru c revision** 。
5. cards 2 to 5 hybrid : 告知系統目前使用的是單倍密度的 hybrid 卡，安裝在第 2 至第 5 個槽；如果加上 double density 則表示這是雙倍密度的 hybrid 卡。同時要在敘述中加上 standard (6Mhz) ， advanced (12Mhz) 及 accuracy (20Mhz) 來宣告使用的 hybrid 卡的測試速度。
(**Note** : 這個敘述要和 target 3070 standard (advanced 或 accuracy) 的敘述相符合才行)
例如: cards 2 to 5 hybrid standard double density 。
6. cards 6 control : 告知系統目前使用的是那個版本的 control 卡。如果使用的是 control XT 卡，則敘述要改為 cards 6 control XT 。
7. supplies hp6624 1 to 4 asru channels 1 to 4 : 告知系統軟體 DUT Power Supply 上所提供的電源經由那些埠 (port) 輸出。
8. probe , debug port : 系統上有一個手持測試用的探棒，用於測試中或是在 debug 的時候。
10. 在每台 Agilent 3070 系統中有一個硬體 config 檔 (H/W config) 存在 /hp3070/diagnostics/th1/config 中，這個檔案所記載的組態必須完全符合實際安裝在系統上的硬體。除此之外，還有軟體 config 檔 (S/W config) 存在各個程式中，例如我們在這個實驗中，我們在 class_bd 這個目錄中寫入了一個軟體 config 檔(S/W config)。在這裡有一件事要特別注意的是，**硬體 config 檔一定要大於或等於軟體 config 檔**，意思是定義在軟體 config 檔中卡片的版本及數目不可以比硬體 config 檔更新或更多。這是因為發展程式時系統會依照每個 board directory 下的軟體 config 檔來分配測試資源(resource)，若 s/w config 定義了不存在的硬體，則系統可能會將這些不存在的資源分配出去，而使得程式與治具無法在本機台上使用。



實驗二 使用線上手冊

1. 啓動 Agilent 3070 線上手冊。

____ 找到 Agilent 3070 線上手冊的書本圖示，按一下往上的箭頭符號將會得到數個不同的說明輔助。

Agilent 3070 APPLICATION NOTES：這是 Agilent 3070 程式應用常見問題。

Agilent 3070 SERVICE MANUALS：介紹 Agilent 3070 硬體的手冊。

Agilent 3070 USER MANUALS：介紹 Agilent 3070 軟體的手冊。

Agilent 3070 FLASH MANUAL：介紹 Agilent 3070 Flash Programming 的手冊。

____ 按 Agilent 3070 USER MANUAL 的圖示：

會出現一個 bookshelf 的視窗,這個視窗分成 4 部份,從左到右分別是,使用 Agilent 3070 (using the Agilent 3070), 系統管理(Administration Topics), 參考資料(Reference), 新增資訊(Additional Information), 在每個主題下面都有相當詳細的資料可供參考, 如以下範例：

2. 查詢 Board Consultant：

____ 在 bookshelf 中”using the Agilent 3070”部份，按下 Test Development Tools

____ 按下第二章 Agilent Board Consultant

這個章節介紹詳細的 Board Consultant 項目，如果有興趣可以看看。

另外還有 INDEX 和 Syntax Reference 可以利用，INDEX 主要是用來查詢關鍵字，Syntax Reference 主要用來查詢指令。

3. 使用 INDEX：

例如：查詢關鍵字 config 的資訊

____ 按下 bookshelf 中的 INDEX

此時會出現 INDEX 視窗，上面有依字母順序排列的列單。

____ 按下英文字母”C”

會直接跳至以”C”為開頭的資料，你可能會使用在螢幕上方的左右箭頭按鈕，或是移動捲軸來讀取更多頁面，或使用尋找的功能。



- ___ 在功能表中按下 Edit
- ___ 下拉式功能表中按下 Find...
- ___ 輸入 config
- ___ 按下 Find Again
會跳至跟 config 有關的資料，可能不只一筆，可連續按下 Find Again,直到看到想要的資料。
- ___ 點選字串後面的章節頁數資料(如:C-15,16-53...)
畫面則會進入有關 config 詳細的資料，如果查詢完畢，可以按右上方的 bookshelf，Syntax Reference，Master Index 任何一個，回到開始查詢的起始畫面。
- ___ 按下右上方的 bookshelf，回到開始查詢的起始畫面。

4. 使用 Syntax Reference：

例：“get” 指令查詢：

- ___ 按下 bookshelf 中的 Syntax Reference
此時會出現 INDEX 視窗,上面有依字母順序排列的列單。
- ___ 按下英文字母”G”
會直接跳至以”G”為開頭的資料，你可能會使用在螢幕上方的左右箭頭按鈕，或是移動捲軸來讀取更多頁面。
- ___ 找到 get 後，點選字串後面的章節頁數資料(如:C-15,16-53...)
畫面則會進入有關 get 詳細的資料，如果查詢完畢，可以按右上方的 bookshelf，Syntax Reference，Master Index 任何一個，回到開始查詢的起始畫面。
- ___ 按下右上方的 bookshelf，回到開始查詢的起始畫面。

5. 請練習查下面字串

Fixture Consultant
IPG Consultant
faon
find pins
pins test
ASRU Card



實驗三 Find Pins 及 Verify Node 練習

Find pins 和 verify node 是用來檢查治具的接觸和繞線的指令，是在 debug 治具時常會用到的工具。

將 Agilent 3070 測試機台按照前面的開機程序開機完畢,再開始下面的實驗

1. 把治具放在 Bank2 上。再將 class board 放在治具上。

___ 按下 Agilent 3070 圖示，在 Agilent Pushbutton Debug 上按一下。

___ 輸入你的目錄路徑：`/home/user<#>/class_bd`。

你會看到 Agilent Pushbutton Debug (Board Level) 視窗。

___ 按下 MACROS

___ 按下 SETUP_MACROS

___ 按下 INITIALIZE_ALL

在BT-BASIC視窗中的命令列(command line)輸入

___ `vacuum well a is 2,3` 再按 **Enter** 鍵。(這是用來設定 faon 所要吸下的 module)

2. 在 testhead 控制列中

___ 按下 faon

3. find pins 指令練習

在BT-BASIC視窗輸入

___ `find pins` 按 **Enter** 鍵

放置探棒在class board上的任何一個測試點，並按下F1，你會看到像下面的訊息，從訊息中可以知道現在找到測試點的它所代表的node name。

```
Found (22244), 1 ohms,
```

```
Node: R203-2
```

```
      r203.2
```

```
      r204.2
```



➤ 也可以這樣做:

輸入完 **find pins** 指令後先按 **F7**(自動跟手動切換鍵)，那麼當你用探棒碰觸任何測試點時，系統就會自動回應訊息。

____ 如果結束 **find pins** 指令，請按 **F8**。

4. **verify node** 指令練習

在 **BT-BASIC** 視窗輸入

____ **verify "R206-2"**

以探棒碰觸指定的測試點來檢查該點所接的繞線是否 **ok**，

如果正常會看到如下面 **Node PASSES** 的訊息。

如果有繞線未繞，會看到標示某 **BRC failed**，如：“**BRC(22221) FAILED**”。

如果有短路，會看到如：“**Short found to BRC (22041), 6 ohms, node R315-1**”的訊息。

```
70: Probe R206-2 (22363)
      (22224)
```

```
      r206.2
```

```
      r207.2
```

```
Node PASSES.
```

5. 在 **BT-BASIC** 視窗輸入

____ **verify "L201-1"**

```
30: Probe L201-1 (21562)
```

```
      l201.1
```

```
Node PASSES.
```

跟 **find pins** 一樣，可以按 **F7** 選擇自動回應訊息。

Note : **find pins** 和 **verify node** 不同的地方在 **verify node** 能檢查出繞線的正確性。他會用程式中所指定應該要繞到待測點的 **brc** 來檢查是否有收到訊號，因此若有少繞或多繞都會有清楚的訊息顯示。而 **find pins** 只能將收到訊號之 **brc** 列出，並不會檢查是否有多繞或少繞線。