

Command	add-data	discription
COM	dos command	<p>經由 com port送資料到 UUT端(含enter) 如果要送空白字可以下 COM SPACE (只有一個空白不含enter)</p> <p>COM OFF , COM ON 可開關 COM PORT</p> <p>COM ON 38400 COM2</p> <p>在ON 時可以設定 COM PORT的傳輸率和指定那一個com port 輸出入</p> <p>COM = 字串變數名稱</p> <p>程式中有二個COM Device可設定,不指定則使用第一個COM Device 設定可由 COM ON/OFF的過程來指定使用 @@1 or @@2 來區隔</p> <p>COM OFF @@2 COM ON 38400 COM2 @@2</p> <p>COM = 字串變數名稱 @@2</p> <p>COM ABCD.EXE @@2</p>
COMFILE	data	<p>經由 com port送TXT檔案資料到 UUT端,但必需COM PORT要先打開,才可以執行這個命令, DATA的字串分成二個部份用 , 號分隔.</p> <p>第一個部份:為每一行之間要延遲多少 ms</p> <p>第二個部份:為所要傳輸的txt file 的路徑和檔名</p> <p>COMFILE 50,D:\123.TXT</p>
COMHEX	ASCII-CODE	<p>可經由 COM PORT送16進位的字串到UUT端.</p> <p>COMHEX 313A00</p> <p>程式中有二個COM Device可設定,不指定則使用第一個COM Device 設定可由 COM ON/OFF的過程來指定使用 @1 or @2 來區隔</p> <p>COMHEX 313A00 @2</p>
GETHEX	ON   OFF	<p>可經由 COM PORT 收進來的資料以16進位顯示,控制以ON OFF 來控制</p> <p>GETHEX ON 表示將COM PORT的資料改以16進位值收入</p> <p>GETHEX OFF 表示將COM PORT的資料以字串形式收入</p>
DIO	name value	<p>參考 DIO.INI [dio] setion &amp; name==&gt;</p> <p>DIO DATA1 255</p> <p>name 可在DIO.INI中自行設定所需的資料</p> <p>如果只有一個bit也可用 DIO data1 ON ; DIO data1 OFF 來做開關的動作</p>

readmec.txt

DIOMACRO	name	參考DIO.INI [DIOMICRO] Setion & name可在DIO. INI中自行設定所需的資料name DIOMICRO POWERON
DIOREAD	name data	參考DIO.INI [dio] setion-name 設定name & 比較資料data 如果NI DIO 取回的資料和data  相同則TRUE 否則回傳 FALSE 可以存到變數中 來判斷 DIOREAD DATA1 = ABCD再以 IF ( ABCD > 234 ) THEN @ERROR1 ELSE @ERROR2  DIOREAD DATA1 = ABCD >> 則會將資料也存到VAR. INI中的savevar section 的變數資料,最後會加到 testlog.txtvar之中
ACHV	name	參考 DIO.INI [ACHV] setion & name==>ACHV V3P7 name 可在DIO. INI中自行設定所需的資料如果 有等號則會存到var. ini中的實數變數中,不判 斷對錯 ACHV TEST1 = MYTEST 結果會存到 VAR. INI中的MYTEST變數中  ACHV TEST1 = MYTEST >> 則會將資料也存到VAR. INI中的savevar section 的變數資料,最後會加到 testlog.txtvar之中,不判斷對錯  ACHV TEST1 >> 則會將資料也存到VAR. INI 中的savevar section的變數資料用errorcode 為名稱,最後會加到testlog.txtvar之中,會 判斷對錯
ACHI	name	參考 DIO.INI [ACHI] setion & name==> ACHI OperationChargeTest1 如果有等號則會存到var. ini中的實數變數中, 不判斷對錯 ACHI TEST1 = MYTEST 結果會存到VAR. INI中的MYTEST變數中  ACHI TEST1 = MYTEST >> 則會將資料也存到VAR. INI中的savevar section 的變數資料,最後會加到 testlog.txtvar 之中,不判斷對錯  ACHI TEST1 >> 則會將資料也存到VAR. INI 中的savevar section 的變數資料用 errorcode為名稱,最後會加到 testlog.txtvar之中,會判斷對錯

SAVEWAIT	ON / OFF	<p>readmec.txt</p> <p>可以設定是否要將等到的字串行存在變數中,以供解析字串使用,需要配合WAIT來使用</p> <p>SAVEWAIT ON 打開等待字串行存檔</p> <p>WAIT GOOD = MYDATA 會將等到GOOD字串的那一行存到VARINI中的varstring</p> <p>變數名稱為 MYDATA 的資料中</p> <p>SAVEWAIT OFF 關掉等待字串行的存檔</p>
WAIT	waitstring,timeout	<p>等待COM port送過來的資料核對,timeout等待時間(1000)表1秒 Default 是3分鐘:</p> <p>例: WAIT C:&gt;,1000</p> <p>WAIT (crc32 is 0xAB744A4C~crc32 is 0x87F536A4)</p> <p>可以等待多個字串</p> <p>~ 表or, &amp; 表 and</p> <p>當等待字串有 , 時可用下面例題設定</p> <p>WAIT (crc32,4,),2000 這時候後面一定要設TIME OUT 的時間,不能設=和&gt;&gt;</p> <p>WAIT crc32 is 0xAB744A4C = TEST,2000 會將找到或找不到寫到字串變數</p> <p>TEST 中 TEST= TRUE or TEST= FALSE可用IF來做判斷在有 = 號的狀態下等到時間超過時不會停止,只會設定變數為FALSE然後繼續往下測試</p> <p>WAIT ABC = TEST &gt;&gt;,2000 WAIT ABC &gt;&gt;,2000</p> <p>沒有= 號的錯誤時直接停止, &gt;&gt; 則會將資料也存到VAR.INI中的savevar section 的變數</p> <p>資料最後會加到testlog.txtvar之中</p> <p>WAIT 後不接任何字串,則UUT傳來的字串會暫時保存在BUFFER中不送出來處理等到下一個</p> <p>WAIT waitstring 字串的command出現時才會將之前的字串送出用以防止大量快速資料處理速度的問題</p>
WAITFILE	waitstring,timeout	<p>是等待某個目錄內是否有waitstring的檔案名稱出現,timeout等待時間(1000)表1秒Default 是3分鐘:</p> <p>例: WAITFILE C:\TEMP\TEST.TXT,1000</p> <p>表示等TEST.TXT這個檔案一秒鐘,超過時間就FAIL,如果要以這個通知錯誤,可用</p> <p>UUT_FAIL.DAT</p> <p>來告知其失敗:</p> <p>要放在同宣告的相同路徑中C:\TEMP\</p>

WAITNOW	readmec.txt	
	waitstring,command,type	<p>等待COM port,Lan 送過來的資料核對,如果存在所收到的字串,則由type所定義的傳輸方法,將字串回傳回去,會一直持續傳送直到下 OFF 為止</p> <p>type: 1,2表com port 傳輸, 3表 Lan 傳輸</p> <p>例: WAITNOW {,d,1  =&gt; 等到 { 字串, 使用 com port 送出 d 這個字元</p> <p>例: WAITNOW {,d@2,1  =&gt; 等到 { 字串, 使用 com2 port 送出 d 這個字元</p> <p>在後面需要等到某個字串收到後要下</p> <p>WAITNOW OFF 才會關掉字串持續的傳送</p>
SLEEP	delay-time	delay ms times => SLEEP 1000 (1000 = 1 second)不作任何事
SMB	name	設定usi smart battery borad電源 reference DIO.INI [ACHI] setion & name => SMB OperaTionChargeTest1
KEY	data	由dio介面送 key值到 UUT 鍵盤 => KEY A 不含ENTER,參考DIO.INI [KEY]鍵值 和 [DIO] KEYDATA 的DIO port 設定
KEYSTRING	data	由dio介面送字串到 UUT keyboard => KEYSTRING test? (?is enter) 其餘設定同 KEY的資料 透過dio介面送資料,一個字元一個字元的傳送
DAC	name	設定 DAC資料參考DIO.INI [DAC] setion & name => DAC SINWAVE ON (ON   OFF),即NI analog 輸出設定
PULSE	NAME ON/ OFF	設定COUNTER 輸出方波訊號,可設定所需的頻率,設定資料參考 DIO.INI [PULSE] setion => PULSE P200 ON 輸出名稱為P200的頻率. PULSE P200 OFF 結束關閉P200的輸出

readmec.txt

ALLKEYTEST	delay-time	可以修改 DIO.INI [ALLKEY] setion 的值來產生輸入順序
POST	data,timeout	經由LPT PORT接收等待post code data ==> POST A0,3000 timeout等待時間(1000)表1秒Default 是3分鐘
LPTW	data	寫資料到LPT PORT,分為二種格式,寫Line and 寫 port port => 0:data port; 1:Status port 2:control port 要寫line則要有三個資料: LPTW port,line,data ==> LPTW 0,2,1 data只能為0,1 寫 port 則只有二個資料 LPTW port,data ==> LPTW 0,234  data < 255
LPTR	data = varName	寫資料到LPT PORT,分為二種格式,寫Line and寫 port port => 0:data port; 1:Status port 2:control port 要寫line則要有三個資料: LPTR port,line = varName ==> LPTR 0,2 = MYTEST Line只能為0-7 寫 port 則只有二個資料 LPTR port = varName ==> LPTW 0 = mydata 會存到變數mydata中可以再配合varrange來判斷對錯或著用if-then-else來決定  目前只支援EPP MODE,請先將BIOS的設定改成EPP MODE 這個command才能用
RJ1145	NON	經由 DIO.INI [RJDATA] setion 的設定,要配合function Board來產生 loop back 功能,用在IBM UUT
END	NON	測試結束時要加這個資料,才能得到測試完成的結果 例 ==> END
ERRORCODE	data	如果UUT端沒有定義錯誤碼,可自行設定測試流程中的錯誤碼 ==> ERRORCODE 0001, UUT端可透過COM port送error code到HOST格式如後ERROR-CODE:0001,ERROR-CODE:會被忽略只保留 0001 如果加 > 號後面的字將會在 buglist中出現

readmec.txt

其錯誤的資訊

ERRORCODE 0001 > TEST FIRST

SENDBARCODE	NON	透過NI DIO 將Barcode經由 UUT keyboard 送到 UUT 端 ==> SENDBARCODE
YESNO	data	設定message box 讓使用者確認 YES OR NO, 按YES會送'YES'到 COM port並且繼續執行下面的指令,按NO 會送'NO'到COM port並且顯示錯誤碼,停止執行其它指令,data為顯示字串  YESNO = ABC 可單純顯示變數字串,但需存在 VAR.INI之中 如不存在則會顯示 ABC: OK
SELECT	data	設定 select box 讓 USER 選擇要使用那一個,USER 選擇後會依選的順序存到 VAR.INI 中的[VARSTRING] 中的 SELECT 變數中要顯示的項目用 ' , ' 來分隔,依順序會存成 1,2,3...的值 SELECT test1,test2,test3 會產生三個選項 test1~3
CLOSESMB	NON	關閉所有smart battery board的電源設定
COUNTER	name	參考 DIO.INI [COUNTER] setion & name ==>COUNTER VGAMODE 主要在量取全波和半波的比值,由上下限來判斷正確與否 COUNTER VGAMODE = MYTEST 結果會存到 VAR. INI中的MYTEST變數中 COUNTER VGAMODE = MYTEST >> 則會將資料也存到VAR.INI中的savevar section 的變數資料,最後會加到 testlog.txtvar之中,不判斷對錯  COUNTER VGAMODE >> 則會將資料也存到 VAR. INI中的savevar section的變數資料用 errorcode為名稱,最後會加到 testlog.txtvar之中,會判斷對錯
SENDMAC	Dos-command	經由 barcode.ini取得 MAC data , Dos-command 是UUT端要寫入MAC資料的 DOS 指令(by chipset support),指令會透過com port結合Dos-command 和MAC 資料來執行,如果COMMAND要和MAC連在一起則下 Command~

readmec.txt

則指令會成為CommandMAC不然是成為  
Command MAC

SENDGUID Dos-command

經由 barcode.ini取得 GUID data ,  
Dos-command 是UUT端要寫入GUID資料的 DOS  
指令(by chipset support),指令會透過com  
port 結合Dos-command 和GUID 資料來執行,  
如果COMMAND要和GUID連在一起則下Command~  
則指令會成為CommandGUID 不然是成為  
Command GUID

WRITEBARCODE Dos-command

經由 barcode.ini取得 BARCODE data ,  
Dos-command 是UUT端要寫入BARCODE 資料的  
DOS指令(by chipset support),指令會透過  
com port結合Dos-command 和BARCODE 資料  
來執行,如果COMMAND要和BARCODE連在一起則  
下 Command~ 則指令會成為CommandBARCODE  
不然是成為  
Command BARCODE  
exp:  
WRITEBARCODE UUT /A/B ,2,5  
==> UUT /A/B  
為UUT 端的命令,而,2,5表示barcode 由第  
二位取長度5來送就會成為 UUT /A/B 23456  
送出去

FAILSTOP

由FCTCOMMAND 主動呼叫的fail stop.沒有參  
數

\*\*\*\*\* GPIB FUNCTION\*\*\*\*\*

POWERSAFE ON/OFF

設定ON 時,一但電源測試錯誤立即斷電,OFF時  
可以重試,default為OFF

GPIBCONFIG name

經由 DIO.INI [GPIBConfig] setion & name  
==>GPIBConfig AAA  
透過GPIB來設定 PowerSupply或GPIB介面的設  
備

SETHPPOWER name

經由DIO.INI [HPPOWERSET] setion & name  
資料來設定GPIB  
PowerSupply ==>SETHPPOWER SET37V

READGPIBDATA name

經由 DIO.INI [GPIBDATA] setion & name的  
資料來設定比較值==>READGPIBDATA

HPCurrent,

可借由TYPE 選擇是要讀取電壓或電流值再與  
UP & LOW 值比較後回傳TRUE OR FASLE  
READGPIBDATA HPCurrent = ABC  
可將回傳的資料存到變數中

readmec.txt

READGPIBDATA HPCurrent = ABC >>

則會將資料也存到VAR.INI中的  
savevar section 的變數資料,最後會加到  
testlog.txtvar之中,不判斷對錯

READGPIBDATA HPCurrent >> 會自行判斷對  
錯,會將資料存到VAR.INI中的  
savevar section 的變數資料用errorcode為  
名稱,最後會加到testlog.txtvar之中會判斷  
對錯

HPOUTPUT            ON/ OFF

經由 GPIB介面設定POWER 輸出是ON or OFF  
==> HPOUTPUT ON

GPIBSTR            DATA

DATA字串透過GPIB PORT來送到GPIB DEVICE

GPIBSTR = 字串變數名稱  
可借由變數來產生命令列  
VARSTRING MYDATA = '\*IDN?';  
GPIBSTR = MYDATA

GETGPIBSTR        DATA = varName

DATA為要對GPIB PORT下達要回傳資料的  
COMMAND,需給變數參數名稱  
GETGPIBSTR CURR:STATE = ABC  
等號前面為DEVICE COMMAND,後面為要  
存的變數名稱ABC  
GETGPIBSTR CURR:STATE = ABC >>  
則會將資料也存到VAR.INI中的  
savevar section 的變數資料,最後會加到  
testlog.txtvar之中,不判斷對錯

\*\*\*\*\* VISA FUNCTION\*\*\*\*\*

VISA Command DATA

主要是透過NI VISA介面來控制所有可由VISA  
存取的界面,如COM ,GPIB,USB...等等

Command 區分為  
1.FindResource  
2.RsrcName  
3.OPEN  
4.CLOSE  
5.WRITE  
6.READ

1. FindResource 為尋找本機是否有相關可連線的 VISA Device,會將所找到的資源放到  
LOG 中,以供判斷是否要做後續控制 VISA的動作.

VISA FindResource

2. RsrcName DATA 為所找到的 resource 字串設定,其為指定VISA Device 使用  
,Data要設定成所要連結的sorce 字串,一定要先設定之後才能



readmec.txt

使用 OPEN CLOSE, WRITE, READ等命令,不然無法使用以下四個命令

VISA RsrcName USB0::0x0957::0x0507::MY44003692::INSTR  
VISA RsrcName ASRL3::INSTR  
VISA RsrcName = DATA 表示由VARSTRING中找出名為DATA的字串來替代

3. OPEN 為打開所指定的設備,沒有 DATA 參數

VISA OPEN

4. CLOSE 為關閉所指定的設備,沒有 DATA 參數

VISA CLOSE

5. WRITE DATA 為依據所打開的 Device 使用手冊下達其相關的指令

VISA WRITE \*IDN?

6. READ varName 需配合 WRITE 來操作,沒有WRITE 不會有READ的資料可讀  
VISA PORT 下達要回傳資料的COMMAND,需給變數參數名稱 varName  
VISA READ = ABC 後面為要存的變數名稱ABC,會存入VARREAL 變數中,所以一定要是實數資料如果不是實數資料要配合SAVEWAIT來做

VISA READ = ABC >> 則會將資料也存到VAR.INI中的savevar section的變數資料,最後會加到testlog.txtvar之中,不判斷對錯

\*\*\*\*\*

ACHWAVE                      name                      經由 DIO.INI [ACHWAVE]取得ACH wave 高點和低點差值是否在top<->low的區間內  
=> ACHWAVE AUDIO  
可存到變數名稱中 ACHWAVE AUDIO = ABCD  
ACHWAVE AUDIO = ABC >>  
則會將資料也存到VAR.INI中的savevar section 的變數資料,最後會加到testlog.txtvar之中,不判斷對錯

ACHWAVE AUDIO    >> 會自行判斷對錯,會將資料存到VAR.INI中的savevar section 的變數資料用errorcode為名稱,最後會加到testlog.txtvar之中

ACHFREQ                      name                      經由 DIO.INI [ACHFREQ]的設定可以判斷 頻率,offset, 峰對峰的值差,次頻率傅利頁轉

readmec.txt

換的大小值

ACHREQ FEI

可存到變數名稱中 ACHREQ FEI = ABCD 會自動以ABCD產生四組資料

ABCD1 = 峰對峰的值差

ABCD2 = OFFSET

ABCD3 = 頻率

ABCD4 = 次頻率的傅利葉轉換的值

ACHREQ FEI = ABCD >> ,不判斷對錯,會將資料也存到VAR. INI中的  
savevar section的變數資料,最後會加到  
testlog.txtvar之中但只存二個值Offset&頻率

ABCD0 = OFFSET

ABCD F = 頻率

ACHREQ FEI >>會自行判斷對錯,會將資料存到VAR. INI中的savevar section 的變數資料  
用errorcode為名稱,最後會加到  
testlog.txtvar之中但只存二個值 Offset & 頻率

errorcode0 = OFFSET

errorcodeF = 頻率

DACPOINT            name    ON/OFF

經由 DIO.INI [DACPOINT] setion & name來  
設定固定的DAC輸出電壓

==> DACPOINT V3P7 ON /OFF

也可以使用VARREAL的變數來設定其值

DACPOINT V3P7 = MYDATA

MYDATA是實數變數,可在設定前給定

例如 VARREAL MYDATA = 3.55

//以下為直接控制NI CARD DIO  
NIDIO            name    value

參考 DIO.INI [dio] setion & name

==>NIDIO DATA1 255

name 可在DIO. INI中自行設定所需的資料

NIDIOREAD            name    data

參考DIO.INI [dio] setion-name 設定name  
& 比較資料data 如果NI DIO 取回的資料和  
data 相同則 TRUE 否則回傳 FALSE

NIDIOREAD MYDIO 233

可以存到變數中來判斷

NIDIOREAD name = ABCD

再以

IF ( ABCD > 234 ) THEN @ERROR1 ELSE @ERROR2

readmec.txt

NIDIOREAD name = ABC >>  
則會將資料也存到VAR.INI中的  
savevar section的變數資料,最後會加到  
testlog.txtvar之中,不判斷對錯

NIDIOREAD name DATA1 >> 會自行判斷對錯  
,會將資料存到VAR.INI中的savevar section  
的變數資料用errorcode為名稱,最後會加到  
testlog.txtvar之中

NIDIODIR            name            參考 DIO.INI [NIDIODIRECT] setion &  
name 可以直接控制NI CARD的IN OUT  
的方向 --->  
NIDIODIR PORT0  
NIDIODIR PORT1  
NIDIODIR PORT3

/\*                                    大範圍的mark,其功能同c++的功能,需配合 \*/  
作為結束

RUNPCEXE            DATA            DATA為執行檔在pc端的路徑,檔案名稱,如果  
該程式已經存在可配合Pfindexe.exe來找程  
式來執行可避免重複執行  
RUNPCEXE Pfindexe.exe 要執行的exe 其  
程式的caption  
==>RUNPCEXE Pfindexe.exe C:\WINDOWS\system32\calc.exe "小算盤"  
==> RUNPCEXE = ABC    則會到 var.ini找字串變數ABC來代入執行

EXECLOSE            DATA            如果以Data為 caption 的視窗存在的話,將其  
程式關閉

RUNSTR              DATA            DATA有這個參數是存在com.INI中的IMAGE  
section中的名稱字串程式會主動到Com.ini  
中找到 SYMBOL section 的APPATH名稱將  
APPATH 和 DATA所指的資料結合變成所需要  
的版本來執行 RUNSTR Monitor會產生執行  
D:\ibmbeta\GetWinH\Multi-Flash\MultiLoad.exe (APPATH)  
d:\MC3000\DVT\BSP4.1.002\Hex Images\MC3000C42B\MC30XXC42XMon0115.bin  
(Monitor)

IF                                    每一個數之間要有空白,字串前後要加 ' 號  
=> 'abcd' 要有跳  
的位置 THEN @abc ELSE @123 運算符號間  
要有空白區隔只能往後跳,不能往前跳,要往  
前跳要配合JUMP來用  
IF ( BOX1 = 'ABC' ) THEN @ABC ELSE @ERROR  
字串比對BOX1是字串變數  
IF ( BOX1 = BOX2 ) THEN @ABC ELSE @ERROR  
字串比對BOX1,BOX2是字串變數

readmec.txt

IF ( ( USB5V\_1 > 4.75 ) AND ( USB5V\_1 < 5.25 ) ) THEN @USB5V\_11

@NAME                      Label 給jump 時使用

JUMP                      Label                      JUMP @123 跳到所標示的位置

\*\*\*\*\* 各各變數名稱宣告不能相同\*\*\*\*\*

LAB                      Source    Distiation                      這個功能在計算color LAB的 Delta E,回傳的資訊會放在 VARREAL LABTEMP中,以供後續判斷Source 是在DIO.INI section [LAB]中的名稱,其存放中間值的 L A B 以空白區隔 Distiation 是以 VARSTRING 所組合出來的L A B字串變數以空白區隔

VARREAL                      DATA                      可事先宣告實數變數,或用來做實數的計算  
VARREAL ABC,DEF  
VARREAL HILL = DEF + (ABC -1) / 2  
用於運算時變數必需先前宣告過,但結果變數HILL則不必可做16進位運算  
VARREAL TEST1 = 0x1101  
VARREAL TEST2 = 0x101  
VARREAL TEST3 = TEST1 AND TEST2  
  
可以使用動態變數名稱宣告  
TEST1 是VARSTRING 或著 VARREAL的內存值  
例如 VARSTRING TEST1 = MYVAR + '1'

VARREAL TEST1 << DEF + (ABC -1) / 2  
則上一行式子會存到 VARREAL 變數中以MYVAR1 為名稱存檔

VARSTRING                      DATA                      可事先宣告字串變數,或用來做字串的運算  
VARSTRING ABC,DEF  
可存字串變數  
VARSTRING ABC = DEF + '1234567'  
其結果為 字串變數DEF加上1234567的結果存到ABC的變數字串中,如果DEF不存在則是給空字串  
字串可以使用動態變數名稱宣告  
TEST1 是VARSTRING 或著 VARREAL的內存值  
例如 VARSTRING TEST1 = MYVAR + '1'

VARSTRING TEST1 << DEF + '1234567'  
則上一行式子會存到 VARSTRING 變數中以MYVAR1 為名稱存檔

如果需要加號可用#來取代 + 例:  
VARSTRING DATA = 'ABC# 234'  
則 ECHO = DATA 會顯示出 ABC+ 234

VARRANGE	NAME	readmec.txt	
		<p>可用於由先前存的變數值和 DIO.INI 中 [VARRANGE] 中的值做比對,看是否在範圍之內但名稱必須要相同才能比較</p> <p>VARRANGE ABC</p> <p>必須在DIO.INI和VAR.INI中都存在 ABC這個變數才能比較</p> <p>VARRANGE ABC &gt;&gt;會將資料存到VAR.INI中的 savevar section 的變數資料以變數名稱為存檔名稱</p>	
COMVAR	VARNAME	LINE	<p>可將 COM OR LAN 傳回來的資料存到VARREAL 的變數中,如果為數字則可用VARRANGE 來做判斷,如果為字串則要使用 IF THEN ELS 的 COMMAND 來判斷 LINE 為要抓傳回的字串第幾行字串存到 VARNAME 所指定的變數中,此時如使用 WAIT 等待字串則會無法使用,必需要等到</p> <p>**SAVE COMVAR OFF**</p> <p>才會恢復正常</p> <p>COMVAR MYTEST 2</p> <p>抓回傳的第二行字串存到變數中</p> <p>COMVAR MYTEST</p> <p>抓回傳的第一行字串存到變數中</p>
SUBSTRING	TYPE	NAME	Position Token
		<p>可將VAR.INI中的 VARREAL or VARSTRING的字串截取出來再存到VAR.INI中</p> <p>1.TYPE = R   S R:表VARREAL資料 S:表VARSTRING的資料</p> <p>2.NAME 表在VAR.INI VARREAL 或著 VARSTRING的變數名稱</p> <p>3.Position 表示要存取的字串所在的位置</p> <p>4.Token 表示用這個字串當分隔字串,不填則當成以空白字串來分隔</p> <p>SUBSTRING R ABC 3 #</p> <p>==&gt;表示 VARREAL 中的ABC 變數以# 來分隔取第三個</p> <p>SUBSTRING S ABC 3 #</p> <p>==&gt;表示 VASTRING 中的ABC 變數以# 來分隔取第三個</p> <p>SUBSTRING S ABC 3 # = TEST</p> <p>==&gt;表示 VASTRING 中的ABC 變數以# 來分隔取第三個,然後存到 TEST變數中</p> <p>SUBSTRING S ABC 3 = TEST</p> <p>==&gt;表示 VASTRING 中的ABC 變數以'空白 來分隔取第三個,然後存到 TEST變數中</p>	
SUBSTRING2	TYPE	NAME	Start Length
		<p>可將VAR.INI中的 VARREAL or VARSTRING的字串截取出來再存到VAR.INI中</p>	

readmec.txt

- 1.TYPE = R | S R:表VARREAL資料  
S:表VARSTRING的資料
- 2.NAME 表在VAR.INI VARREAL  
或著 VARSTRING的變數名稱
- 3.Start 表示要存取的字串起始位置
- 4.Length 表示從起始位置取多長的資料

SUBSTRING R ABC 3 3

==>表示 VARREAL 中的ABC 變數從3開始取長度3

SUBSTRING S ABC 3 3

==>表示 VASTRING 中的ABC 變數從3開始取長度3

SUBSTRING S ABC 3 3 = TEST

==>表示 VASTRING 中的ABC 變數從3開始取長度3然後存到 TEST變數中

SUBSTRING R ABC 3

==>表示 VARREAL 中的ABC 變數從3開始取到結尾

SAVEVARTXT TYPE\*NAME,...

可將變數存到一個檔案之中,可於程式中設定,在結束後會存放在TEST LOG的最後面  
TYPE\*NAME, TYPE\*NAME, .....

1.TYPE\*NAME, TYPE\*NAME, .....

R:表VARREAL資料

S:表VARSTRING的資料)

SAVEVARTXT R\*DATA1,S\*DATA2,R\*DATA3

GetINIData Section name TYPE

可由symbol.ini 中的section 和 name 中取出資料放到 var.ini 中的 VARSTRING or VARREAL 的 section中以相同的名稱 name 來存放

TYPE : R / S R:表示放 VARREAL 中

S:表示放到 VARSTRING中

GetINIData VCC VCC3 R

==>表示從 symbol.ini 中的VCC Section 取名稱為VCC3的值放到 var.ini 中的 VARREAL Section 中名稱為 VCC3

GetINIData VCC VCC3 S

==>表示從 symbol.ini 中的VCC Section取名稱為VCC3的值放到 var.ini 中的 VARSTRING Section 中名稱為 VCC3

SaveINIData Section name TYPE

可由var ini 中的 TYPE 和 name 中取出資

readmec.txt

料放到 symbol.ini 中的 SECTION中以相同  
的名稱 name 來存放

TYPE : R / S

R:表示從 VARREAL 中取出

S:表示從 VARSTRING 中取出

SaveINIData VCC VCC3 R

==>表示從 var.ini中的VARREAL Section

取名稱為VCC3的值放到 symbol.ini 中的

VCC Section 中名稱為 VCC3 的地方

SaveINIData VCC VCC3 S

==>表示從 var.ini中的VARSTRING Section

取名稱為VCC3的值放到 symbol.ini 中的

VCC Section 中名稱為 VCC3 的地方

\*\*\*\*\*

UIMACRO

NAME

可事先於 DIO.INI中的 [UIMACRO] 中定義所  
要的COMMAND,如此可以簡單的命令來下達,可  
參考

INIREADME UIMACRO PORWRON

TIMESTOP

DATA

可設定其多久程式要停止,當其時間到時會將  
VAR.INI中的 字串 TIMESTOP=TURE

DATA ==> ON stoptag 其stoptag 以分為  
計算單位 TIMESTOP ON 3

==>表示計時到三分鐘後結束

TIMESTOP OFF 則即時關掉計時

主要使用在AUTO RUN時做LOOP 要做多久的  
設定要設定AUTO時要設定Barcode.ini中的  
[logfile] 中的 AUTO=TRUE;

ECHO

DATA

可將DATA 的文字顯示到 LOG FILE 中; 如果  
用 ECHO = ABC表示顯示變數值,不存在會顯  
示 NON DATA,如果MESSAGE ON的話,也會顯示  
到其中在字串中加 ~ 會自動換行

ECHO ABCD~ABCD

MESSAGE

ON / OFF

可以將訊息視窗顯示,可透過 ECHO的 Command  
來更改示窗資料

INPUTBOX

CAPTION = VARNAME

可在INPUTBOX中將輸入的字串存到 VARNAME  
的變數中,CAPTION是要顯示的說明字串

INPUTBOX MY BARCODE = BARCODE 則輸入的  
資料會存到VAR.INI中的VARSTRING中的  
BARCODE變數中.

readmec.txt

LAN                      dos command                      經由 lan送資料到 UUT端ON / OFF用來設定 LAN的開關  
LAN ON 4000            / LAN OFF  
在on 時可改變lan 的port 值,在ON的後面加上Port的值即可  
LAN SEND-COMMAND-String  
也可以用 = 後面接變數的方式來傳送  
SAVEBARCODE

VARSTRING FEI = 'C:\ABC\ABC.EXE '+ BARCODE  
LAN = FEI 用字串變數來送參數

TOBARCODEFORM      Type,Data                      經由 TCP/IP 送資料到 Barcode form  
1: 表示所送的資料為 MAC data  
2: 表示所送的資料為 GUID data  
3: 表示所送的資料為 SSN data  
  
4: 表示所送的資料為 SN2 data  
5: 表示所送的資料為 MO\_NUMBER data

TOBARCODEFORM 1,123456789012

也可以用 = 後面接變數的方式來傳送  
VARSTRING FEI = '1,'+ MACCODE  
TOBARCODEFORM = FEI 用字串變數來送參數

1)測試前若 GUID欄位有輸入,會送給 SFIS 作檢查  
2)可用 TOBARCODEFORM 命令直接於 FCTCommand 內編輯送 MAC 或 GUID 及其後之欄位資料給 SFIS 的字串  
(記得要 LAN ON)

Example:

a. 送單一MAC:

TOBARCODEFORM = 1,123456789012

送多個MAC:

VARSTRING PATH = '1,MAC1=0000000000001 MAC2=0000000000002 MAC\_CNT=2'  
TOBARCODEFORM = PATH

b. 只送GUID:

TOBARCODEFORM = 2,1234567890123456

GUID後的欄位也送:(這部份各專案要跟SFIS 討論哪個欄位能填資料,以 D9900 為例, SFIS提供 GUID 後一欄位送其他資料,記得要用逗號隔開)

VARSTRING PATH = '2,0010207318112233,CSN1=000000000000001 CSN2=000000000000002'  
TOBARCODEFORM = PATH

PS. 以上功能適用於 SendBarcodeOnly=OFF  
及 ShopFlow=ON 的情況



readmec.txt

在Barcode form 收到資料後會回傳DATA的值  
可依這個來判斷Barcode from是否收到

CASE	DATA	<p>DATA值為字串變數存在在VAR.INI中的 VARSTRING中可依據DATA來找要JUMP的Lable 如果要組合字串可用 VARSTRING FEI = VAR1 + VAR2 + VAR3 然後用 CASE FEI 如果DATA 的字串變數值不存在則會跳到 @ELSE 所以其後一定要設定一個 @ELSE 的 Lable</p>
SAVEBARCODE	NON	<p>會存在VAR.INI中的 VARSTRING section中, 以變數名稱BARCODE,MAC,GUID的名稱存字串 變數,如果有多個MAC的參數存在 MultiMAC section 會存到MAC1,MAC2...的變數中. 最後SCAN的 Mac code 會在Control中,其它 的會放在MultiMAC 按輸入的順序存放</p>
HOSTIP	DATA	<p>會將HOST IP Address 存到 VAR.INI中的 VARSTRING section中,以變數 DATA 為名稱 存在變數中 ==&gt; HOSTIP MYIP 會將 IP address 存到 MYIP的變數中</p>
GETINI	SECTION NAME VARNAME	<p>會將Prod\symbol.ini 中的 section- name 的資料存到字串變數 VARNAME 之中,以提供 程式中來存取使用,在symbol.ini中主要設 定和local pc相關的變動資料,因為可能每 台pc值會不同,所以單獨設定在此供 fct command 使用, 例如 usb id , com port BT MAC 等等 GETINI MAC 1 MYMAC ==&gt;這段是到 Symbol.ini 中的MAC section ,名稱為 1 的值存到var.ini的字串變數 MYMAC 中</p>
	<p>在SYMBOL.INI 中的設定 [MAC] 1= ABCD12345678 2= TESTABC</p>	<p>如此我們可以將FCTCOMMAND 一致化不用每 台PC要設定不同的流程</p>

SAVELOG	NON	會將目前的LOG 資料先行存檔,不用等到最後按完成才存檔
---------	-----	------------------------------

GETTIME	ON/OFF	當以 GETTIME ON時系統會開始計時,到 GETTIME OFF時會將其間的時間存到 [VARREAL]中GETTIME變數中以ms為單位
---------	--------	--

\*\*\*\*\*

UUT 端可利用 COM PROT 或 LAN 回傳以下列為起始的資料,則 HOST會當成單獨處理資訊

\*\*\*\*\*

ERROR-CODE:0001	會在主程式顯示UUT目前測到那個ERROR 項目
ERROR-UUT:0001	會將主程式的ERROR CODE 改成 0001

UUT-FAIL	會通知主程式測試失敗,做結束的動作
----------	-------------------

VARSTRING ABC = 'TEST DATA'	會將Client端送過去的 TEST DATA 存到 VAR.INI的 VARSTRING中的 ABC中,設定方法可參考 VARSTRING 的 COMMAND
-----------------------------	--

VARREAL ABC = 23	會將Client端送過去的 23 存到 VAR.INI的 VARREAL中的 ABC中,設定方法可參考 VARREAL 的 COMMAND
------------------	---