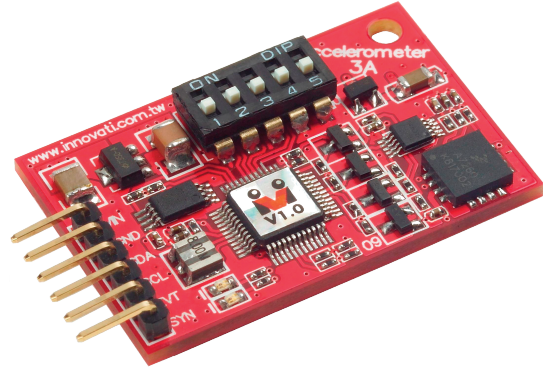


Accelerometer 3A

三軸加速度感測模組

版本: V2.0



產品介紹: 利基 Accelerometer 3A 模組是一個簡單易用的高精確度三軸加速度感測模組，透過 cmdBUS 與 BASIC Commander 連接，可以用簡單的指令取得三個軸向的加速度值，或是加速度與軸向的夾角，並且可執行軟體校正，提升了多種情況的適應性。

應用方向:

- 靜態傾斜角度的量測，控制載具維持平衡。
- 動態加速度的量測，感測受力大小與方向。
- 靜態加速度的量測，可以偵測重力方向。

產品特色:

- 數位化量測三個軸向的加速度值。
- 提供四種精準度(1.5g, 2g, 4g, 6g)，可隨時以軟體指令切換，配合不同量測需求。
- 以度為單位量測加速度與軸向角度值。
- 提供 2D 合成力與角度回傳，並且可任意選擇所要量測的兩個軸向。
- 提供 3D 合成力與角度回傳。
- 受力值提醒，設定所要偵測的受力值，能自動產生提醒事件，可以分別設定 X, Y, Z 或 2D 合成力的受力值，各個設定都有八組提醒事件能彈性運用。
- 偏向角提醒，設定所要比較的基準角與偏移量，以度為單位，當超過偏移角度，自動產生提醒事件，可以設定 2D 任兩軸合成力的偏向角。
- 256 組角度儲存空間，可以儲存現在角度，或是儲存任意角度，做為偏向角設定時的基準。
- 偵測角度可以精準到以度為單位。
- 能偵測範圍達 ± 6 g 的加速度值。
- 自動儲存各軸向最大加速度值，透過指令能快速讀取或重新計算。
- 感測頻率能選擇五種更新速率(100 Hz, 50 Hz, 25 Hz, 10 Hz, 1 Hz)，並能隨時以指令動態更換更新頻率。
- 更新感測值提醒事件，指令設定後，會自動在每次感測值更新後，產生提醒事件。
- 校正設定指令，可以隨時以指令更改校正值，取得符合需求的回傳值。
- 可以直接以指令讀取各軸向的數位電壓值
- 可透過 I2C，下達指令。

連接方式: 直接將 ID 開關撥至欲設定的編號，再將 cmdBUS 連接至 BASIC Commander 上對應的腳位，就可透過 BASIC Commander 執行操作。

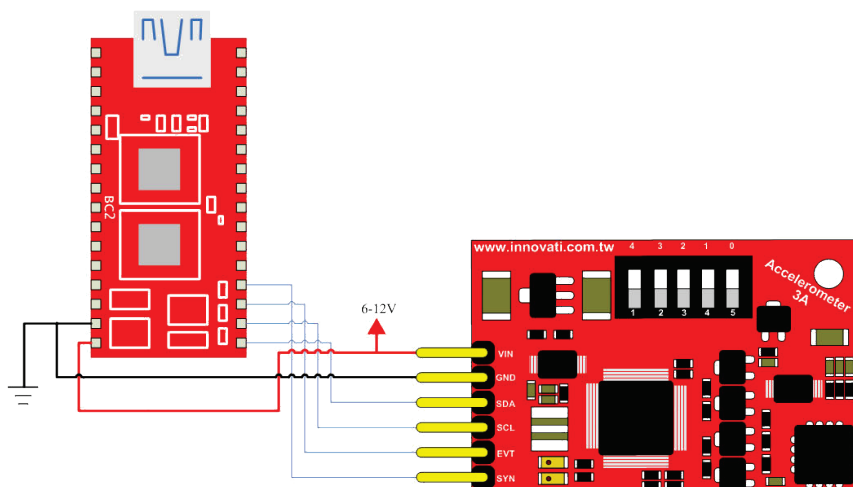
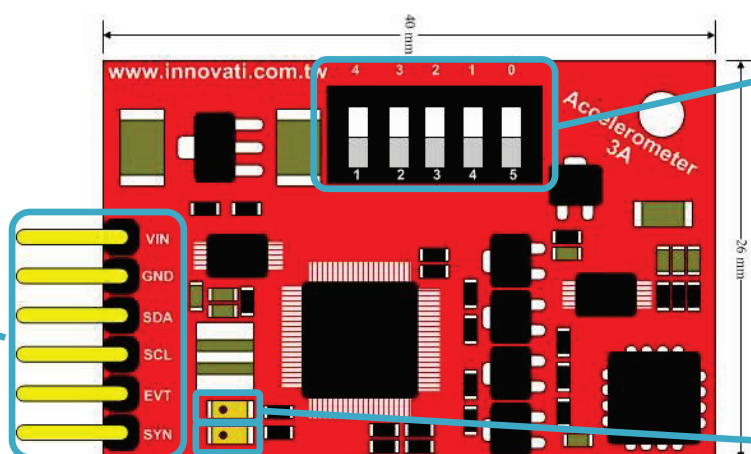


圖 1: 與 BASIC Commander 連接

產品規格:

cmdBUS 接腳，將此處腳位與 BASIC Commander 對應腳位相接，即可透過 BASIC Commander 操控 Accelerometer 3A 模組(連接時請注意腳位對應，將 Vin 對接 BASIC Commander 上的 Vin 腳位，若是腳位錯誤可能造成模組

模組編號設定開關，由右至左以二進制設定 Accelerometer 3A 模組的模組編號，編號可以讓 BASIC Commander 操控時，判斷想要控制的模組(請參考附錄 2)



由上而下依序為:
綠色事件指示燈，閃爍代表模組在傳送事件
橘色指令指示燈，閃爍代表模組與 SBC 正在收送資料

圖 2: 模組腳位與開關介紹

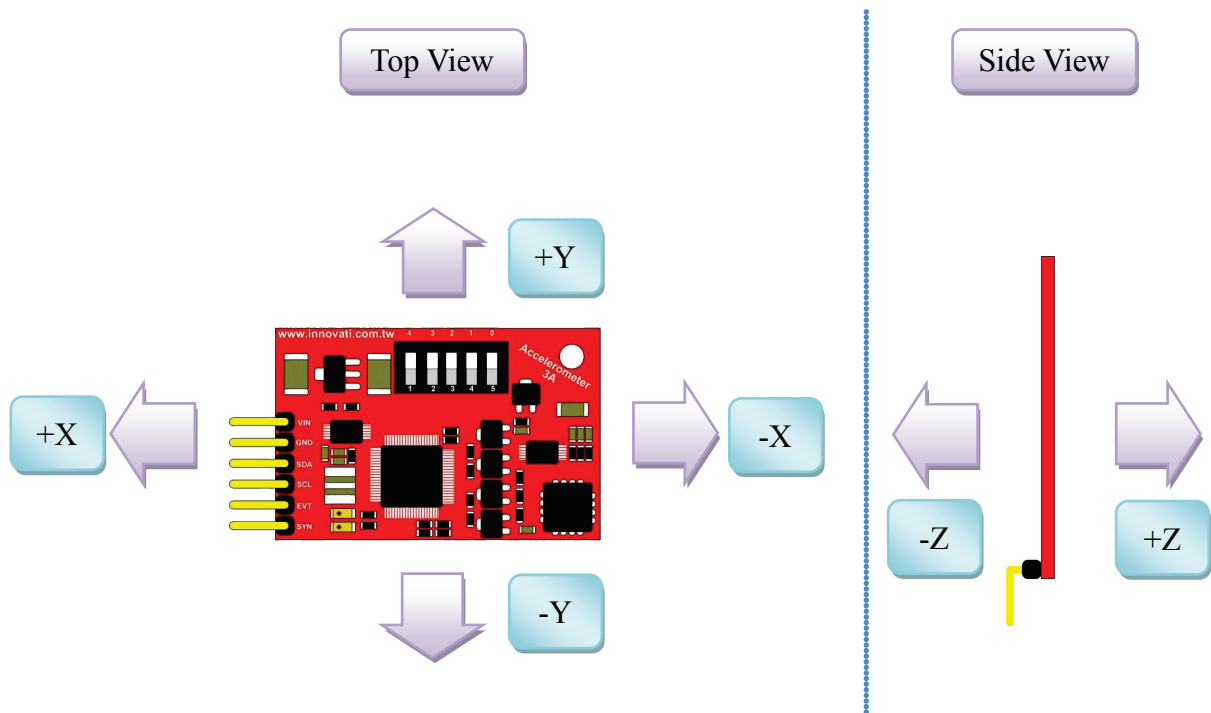


圖 3: 動態加速各軸變化值

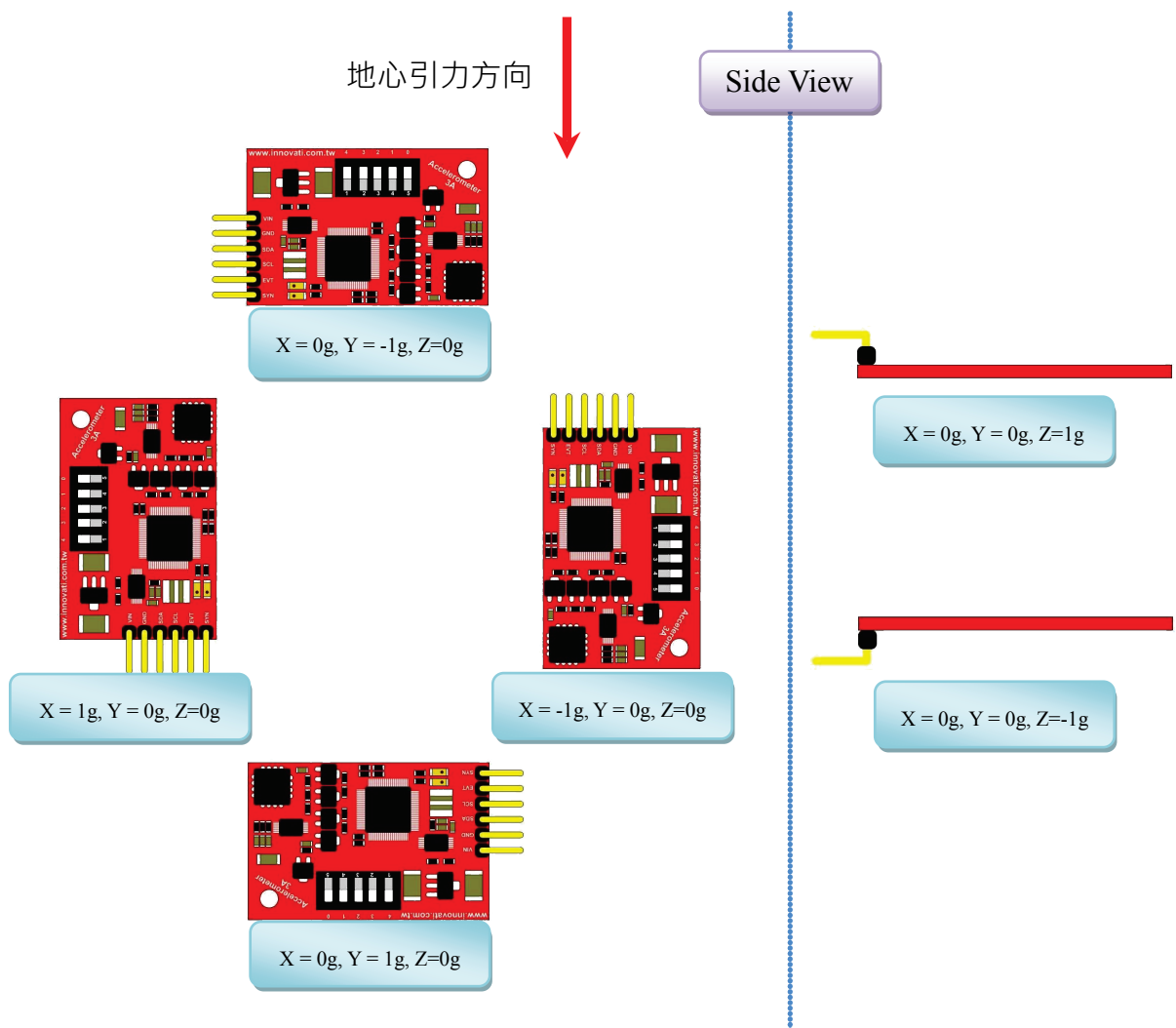


圖 4: 靜態各軸顯示值

操作注意事項:

- 請盡量將模組平放，以獲得較佳的量測值。

模組操作溫度 0 °C ~ 70 °C

模組儲存溫度 -40 °C ~ 125 °C

模組下達指令的方式可分為兩種：**cmdBUS**、**I2C** 控制方式

cmdBUS 指令表:

下面的指令表是專供控制 Accelerometer 3A 模組的各種指令，必要輸入的指令名稱與參數，以粗底或粗斜體表示，粗體的文字在輸入時請不要更改，粗斜體的文字請自行定義適當格式的參數填入。輸入時請注意 innoBASIC Workshop 大寫與小寫會視為相同字。

在執行 Accelerometer 3A 指令前，請先於程式開頭定義對應參數與編號，例:

Peripheral *ModuleName* As Accelerometer3A @ *ModuleID*

I2C 通訊協議(Protocol):

為了使更廣泛的使用者能控制模組，提供了部份指令的通訊協議讓使用者應用。透過通訊規格，使用者可使用 I2C 通訊協議為模組下達命令。

通訊協議常見的封包如下：

MID：模組 ID 編號，空間大小為 Byte 的變數。對應於硬體的指播開關。

CID：命令 ID 編號，空間大小為 Byte 的變數。依不同命令而改變。

Checksum1：驗證位元_1，空間大小為 Byte 的變數。

定義方式： $255 - (MID * 2) - CID$

Checksum2：驗證位元_2，空間大小為 Byte 的變數。

定義方式： $255 - (\text{Checksum1} \sim \text{Checksum2} \text{ 之間的變數總和})$

Checksum3：驗證位元_3，空間大小為 Byte 的變數。

定義方式： $255 - MID - (\text{MID} \sim \text{Checksum3} \text{ 之間的變數總和})$

Dummy：虛設位元，可為任意變數。空間大小為 Byte 的變數。

於通訊規格**每筆資料空間大小階為 Byte**，若資料空間大小超過一個 Byte 時，需將資料拆開，並由 Low Byte 開始傳送。

Ex: 傳送資料 Temp 為一筆空間大小為 Word 的資料，則需將 Temp 拆開，分為 Temp_L、Temp_H，並且**先傳送 Temp_L**。

Ex1 模組編號為 2，命令編號為 153，傳送參數 Byte 為 100，通訊協議為

MID+CID+Checksum1+Byte+Checksum2+Dummy 則：

MID = 2

CID = 153

Checksum1 = $255 - (2 * 2) - 153 = 98$

Byte = 100

Checksum2 = $255 - 100$

Dummy = 0~255 之間的任意數

Ex2 模組編號為 2，命令編號為 153，傳送參數 Temp 為 511，通訊協議為 MID+CID+Checksum1+Temp_L+Temp_H+Checksum2+Dummy 則：

MID = 2

CID = 153

Checksum1 = $255 - (2 \times 2) - 153 = 98$

Temp_L = 255，Temp_H = 1

Checksum2 = $255 - \text{Temp_L} - \text{Temp_H} = 255$

Dummy = 0~255 之間的任意數

指令格式	指令功能
加速度與角度偵測相關指令	
GetXForce(<i>ForceX</i>) GetYForce(<i>ForceY</i>) GetZForce(<i>ForceZ</i>) CmdBUS： GetXYZForce(<i>ForceX</i>, <i>ForceY</i>, <i>ForceZ</i>)	取得 X,Y,Z 軸向的力量值，X 軸向存在 <i>ForceX</i> ，Y 軸向存在 <i>ForceY</i> ，Z 軸向存在 <i>ForceZ</i> ，回傳值為-32768~32767 間的整數值，單位會根據設定模式不同，請參考 SetMode 指令中，各個設定的單位值
I2C： Out： MID+93+Checksum1+Dummy In： MID+ForceX_L+ForceX_H+ForceY_L+ForceY_H+ForceZ_L+ForceZ_H+Checksum3	
GetForce2D(<i>Force</i>, <i>Angle</i>)	取得 2D 平面的合成力量值，存在 <i>Force</i> ，單位會根據設定模式不同，請參考 SetMode 指令中，各個設定的單位值，同時取得力與主軸向的夾角存在 <i>Angle</i> ，單位為度，主軸向的設定可以用 SetAxis2D 指令更改， <i>Force</i> 會回傳 0~65535 間的整數， <i>Angle</i> 會回傳 0~359 間的整數
CmdBUS： GetAngle2D(<i>Angle</i>)	取得 2D 平面的合成力與主軸向的夾角存在 <i>Angle</i> ，單位為度，主軸向的設定可以用 SetAxis2D 指令更改， <i>Angle</i> 會回傳 0~359 間的整數
I2C： Out： MID+95+Checksum1+Dummy In： MID+Angle_L+Angle_H+Checksum3	

GetForce3D(<i>Force</i>, <i>Angle1</i>, <i>Angle2</i>)	取得 3D 的合成力量值，存在 <i>Force</i>，單位會根據設定模式不同，請參考 SetMode 指令中，各個設定的單位值，同時取得於 XY 平面的分力與 X 軸向的夾角存在 <i>Angle1</i>，單位為度，力與 Z 軸的夾角存在 <i>Angle2</i>，單位為度 <i>Force</i> 會回傳 0~65535 間的整數，<i>Angle1</i> 會回傳 0~359 間的整數，<i>Angle2</i> 會回傳 0~179 間的整數
CmdBUS : GetAngle3D(<i>Angle1</i>, <i>Angle2</i>) I2C : Out : MID+97+Checksum1+Dummy In : MID+Angle1_L+Angle1_H+Angle2+Checksum3	取得 3D 的合成力於 XY 平面的分力與 X 軸向的夾角存在 <i>Angle1</i>，單位為度，力與 Z 軸的夾角存在 <i>Angle2</i>，單位為度，<i>Angle1</i> 會回傳 0~359 間的整數，<i>Angle2</i> 會回傳 0~179 間的整數
CmdBUS : GetXADVal(<i>Value</i>) I2C : Out : MID+162+Checksum1+Dummy In : MID+Value_L+Value_H+Checksum3	
CmdBUS : GetYADVal(<i>Value</i>) I2C : Out : MID+163+Checksum1+Dummy In : MID+Value_L+Value_H+Checksum3	
CmdBUS : GetZADVal(<i>Value</i>) I2C : Out : MID+164+Checksum1+Dummy In : MID+Value_L+Value_H+Checksum3	取得各軸向(X，Y，Z 軸)受力的數位化電壓值，存於 <i>Value</i> 中，<i>Value</i> 會回傳 0~65535 間的整數值

2D 偏向角偵測與設定相關指令	
SaveCurrAngle 2D(<i>Number</i>)	將模組現在量測到的 2D 角度，儲存到 <i>Number</i> 指定的位置， <i>Number</i> 可以是 0~255 間的整數
CmdBUS： SaveAngle2D(<i>Number</i> , <i>Angle</i>) I2C： MID+104+Checksum1+ <i>Number</i> + Angle_L+Angle_H+Checksum2+ Dummy	將輸入的 <i>Angle</i> ，儲存在 <i>Number</i> 指定的位置， <i>Number</i> 可以輸入 0~255 間的任意整數， <i>Angle</i> 可以輸入 0~359 間的任意整數
LoadAngle2D(<i>Number</i> , <i>Angle</i>)	讀取 <i>Number</i> 所指定的 2D 角度儲存值，將其放在 <i>Angle</i> 中， <i>Number</i> 可以輸入 0~255 間的任意整數，回傳的 <i>Angle</i> 會是在 0~359 間的整數
CmdBUS： GetDevAngle2D(<i>Number</i> , <i>Angle</i>) I2C： Out： MID+106+Checksum1+ <i>Number</i> + Checksum2+Dummy In： MID+Angle_L+Angle_H+Checksum3	取得與設定為基準方位差異的偏向角。這個指令會把儲存於 <i>Number</i> 位置的角度值，設定為基準方位， <i>Number</i> 請輸入 0~255 間的整數，接著回傳現在量測到的 2D 角度，與基準方向的偏向角，以度為單位，儲存在 <i>Angle</i> 中，如果現在量測的方位，在基準方位逆時針方向 180 度內，回傳值就是正的，如果現在角度在基準角的順時針方向 179 度內，回傳值就是負的， <i>Angle</i> 會回傳-179~180 間的整數
SetDevAngleLimit2D(<i>Angle</i>)	以度為單位，設定判斷 2D 偏向角的差異值， <i>Angle</i> 可以輸入 0~179 間的整數，預設值為 5
GetDevAngleLimit2D(<i>Angle</i>)	回傳設定的 2D 偏向角差異值，以度為單位儲存於 <i>Angle</i> 中，範圍為 0~179 間的整數
SetDevAngleNum2D(<i>Number</i>)	以 <i>Number</i> 編號儲存的 2D 角度值，做為偵測 2D 偏向角的基準方位， <i>Number</i> 可以輸入 0~255 間的整數
GetDevAngleNum2D(<i>Number</i>)	回傳設定的 2D 偏向角基準方位儲存位置，放於 <i>Number</i> 中， <i>Number</i> 會回傳 0~255 間的整數
EnableDevAngleLimitEvent2D()	啟動超出 2D 偏向角極限值提醒事件
DisableDevAngleLimitEvent2D()	關閉超出 2D 偏向角極限值提醒事件
<i>Status</i> = GetDevAngleLimitStatus2D()	回傳現在 2D 角度是否大於 2D 偏向角極限值，若 <i>Status</i> 回傳為 1，則代表現在偵測的方位大於 2D 偏向角極限，若 <i>Status</i> 回傳為 0，則代表在 2D 偏向角極限內

受力提醒設定相關指令	
SetXForceLimit(<i>Number</i>, <i>Limit</i>)	以輸入的 <i>Number</i> 設定要更改的組別，可以輸入 0~7 間的任意整數，並以 <i>Limit</i> 設定受力極限提醒值，可以輸入 0~65535 間的任意整數，只要該軸向(X, Y, Z 軸)受力絕對值超過該值，就會判斷為超過設定
SetYForceLimit(<i>Number</i>, <i>Limit</i>)	
SetZForceLimit(<i>Number</i>, <i>Limit</i>)	
GetXForceLimit(<i>Number</i>, <i>Limit</i>)	以輸入的 <i>Number</i> 取得設定軸向(X, Y, Z 軸)受力極限提醒值，可以輸入 0~7 間的任意整數，並將設定的極限值存於 <i>Limit</i> ， <i>Limit</i> 會回傳 0~65535 間的整數
GetYForceLimit(<i>Number</i>, <i>Limit</i>)	
GetZForceLimit(<i>Number</i>, <i>Limit</i>)	
<i>Status</i> = GetXForceLimitStatus()	取得各組軸向(X, Y, Z 軸)受力極限設定的狀態，存在 <i>Status</i> 中，每個 bit 對應到各組設定，例： <i>Status</i> = 1 ➡ 代表第 0 組設定，到達設定的受力極限，而 1~7 則沒有到達受力極限 <i>Status</i> = 2 ➡ 代表第 1 組設定，到達設定的受力極限，而 0 與 2~7 沒有到達受力極限 <i>Status</i> = 3 ➡ 代表第 0 與第 1 組設定，都已到達設定的受力極限
<i>Status</i> = GetYForceLimitStatus()	
<i>Status</i> = GetZForceLimitStatus()	
EnableXForceLimitEvent()	啟動軸向(X, Y, Z 軸)受力極限提醒
EnableYForceLimitEvent()	
EnableZForceLimitEvent()	
DisableXForceLimitEvent()	關閉軸向(X, Y, Z 軸)受力極限提醒
DisableYForceLimitEvent()	
DisableZForceLimitEvent()	
SetForceLimit2D(<i>Number</i>, <i>Limit</i>)	以輸入的 <i>Number</i> 設定要更改的組別，可以輸入 0~7 間的任意整數，並以 <i>Limit</i> 設定受力極限提醒值，可以輸入 0~65535 間的任意整數，只要設定的 2D 平面受力絕對值超過該值，就會判斷為超過設定
GetForceLimit2D(<i>Number</i>, <i>Limit</i>)	以輸入的 <i>Number</i> 取得設定 2D 平面受力極限提醒值，可以輸入 0~7 間的任意整數，並將設定的極限值存於 <i>Limit</i> ， <i>Limit</i> 會回傳 0~65535 間的整數
<i>Status</i> = GetForceLimit2DStatus()	取得各組 2D 平面受力極限設定的狀態，存在 <i>Status</i> 中，每個 bit 對應到各組設定，例： <i>Status</i> = 1 ➡ 代表第 0 組設定，到達設定的受力極限，而 1~7 則沒有到達受力極限 <i>Status</i> = 2 ➡ 代表第 1 組設定，到達設定的受力極限，而 0 與 2~7 沒有到達受力極限 <i>Status</i> = 3 ➡ 代表第 0 與第 1 組設定，都已到達設定的受力極限

EnableForceLimit2DEvent()	啟動 2D 平面受力極限提醒
DisableForceLimit2DEvent()	關閉 2D 平面受力極限提醒
SetForceLimit3D(<i>Number, Limit</i>)	以輸入的 Number 設定要更改的組別，可以輸入 0~7 間的任意整數，並以 Limit 設定受力極限提醒值，可以輸入 0~65535 間的任意整數，只要設定的 3D 受力絕對值超過該值，就會判斷為超過設定。
GetForceLimit3D(<i>Number, Limit</i>)	以輸入的 Number 取得設定 3D 受力極限提醒值，可以輸入 0~7 間的任意整數，並將設定的極限值存於 Limit ， Limit 會回傳 0~65535 間的整數。
<i>Status</i> = GetForceLimit3DStatus()	取得各組 3D 受力極限設定的狀態，存在 Status 中，每個 bit 對應到各組設定，例： Status = 1 → 代表第 0 組設定，到達設定的受力極限，而 1~7 則沒有到達受力極限 Status = 2 → 代表第 1 組設定，到達設定的受力極限，而 0 與 2~7 沒有到達受力極限 Status = 3 → 代表第 0 與第 1 組設定，都已到達設定的受力極限
EnableForceLimit3DEvent()	啟動 3D 受力極限提醒
DisableForceLimit3DEvent()	關閉 3D 受力極限提醒
受力最大值取得與清除相關指令	
GetMaxXForce(<i>Force</i>)	取得軸向(X, Y, Z 軸)發生過的最大受力值，存在 Force 中， Force 會回傳-32768~32767 之間的整數
GetMaxYForce(<i>Force</i>)	
GetMaxZForce(<i>Force</i>)	
GetMaxForce2D(<i>Force, Angle</i>)	取得 2D 平面發生過的最大受力值，存在 Force 中，以及當時的 2D 角存在 Angle 中， Force 會回傳-32768~32767 之間的整數， Angle 則會回傳 0~359 之間的整數
GetMaxForce3D(<i>Force, Angle1, Angle2</i>)	取得 3D 發生過的最大受力值，存在 Force 中，以及當時的 XY 平面分力與 X 軸夾角存在 Angle1 中，與 Z 軸夾角存在 Angle2 中， Force 會回傳-32768~32767 之間的整數， Angle1 會回傳 0~359 之間的整數， Angle2 會回傳 0~179 之間的整數
ClearMaxXForce()	清除軸向(X, Y, Z 軸)最大受力值的紀錄
ClearMaxYForce()	
ClearMaxZForce()	
ClearMaxForce2D()	清除 2D 平面最大受力值的紀錄
ClearMaxForce3D()	清除 3D 平面最大受力值的紀錄

各項設定相關指令	
CmdBUS : SetMode(<i>Mode</i>)	根據輸入的 Mode 值，設定加速度感測靈敏度，預設值為 0。 Mode 可以輸入 0~3: Mode = 0 → 當量測到 Force = 800 等於 1g，適合量測±1.5 g 內的加速度值 Mode = 1 → 當量測到 Force = 600 等於 1g，適合量測±2 g 內的加速度值 Mode = 2 → 當量測到 Force = 400 等於 1g，適合量測±4 g 內的加速度值 Mode = 3 → 當量測到 Force = 300 等於 1g，適合量測±6 g 內的加速度值
I2C : MID+88+Checksum1+Mode+Checksum2+Dummy	
GetMode(<i>Mode</i>)	回傳設定的感測靈敏度值， Mode 將回傳 0~3 間的整數
CmdBUS : SetRefreshFreq(<i>Rate</i>)	根據輸入的 Rate 值，設定模組量測值的更新速率，預設值為 1。 Rate 可以輸入為 0~4 等五種速率: Rate = 0 → 每 10 ms 更新角度值 (100 Hz) Rate = 1 → 每 20ms 更新角度值 (50 Hz) Rate = 2 → 每 40ms 更新角度值 (25Hz) Rate = 3 → 每 100ms 更新角度值 (10 Hz) Rate = 4 → 每 1000 ms 更新角度值 (1 Hz)
I2C : MID+98+Checksum1+Rate+Checksum2+Dummy	
GetRefreshFreq(<i>Rate</i>)	回傳設定的量測值更新速率，存在 Rate ， Rate 會回傳 0~4 間的整數值，各數字代表的更新速率同 SetRefreshFreq() 的設定
CmdBUS : Status = GetRefreshStatus()	
I2C : Out : MID+100+Checksum1+Dummy In : MID+Status+Checksum3	讀取更新狀態，當 Status 回傳值為 1 時，代表量測值已經被更新過，此時會把內建的狀態值設定為 0，直到模組再次更新量測值，才會設定為 1
EnableRefreshEvent()	啟動量測值更新完成提醒事件
DisableRefreshEvent()	關閉量測值更新完成提醒事件
CmdBUS : SetAxis2D(<i>Type</i>)	以 Type 設定 2D 判斷時的設定軸， Type 可以輸入 0~5，分別代表: 0 → X 為 0 度軸，Y 為 90 度軸 1 → Y 為 0 度軸，X 為 90 度軸 2 → X 為 0 度軸，Z 為 90 度軸 3 → Z 為 0 度軸，X 為 90 度軸 4 → Y 為 0 度軸，Z 為 90 度軸
I2C : MID+170+Checksum1+Type+Checksum2+Dummy	

	5→ Z 為 0 度軸，Y 為 90 度軸
GetAxis2D(<i>Type</i>)	取得 2D 判斷時的設定軸模式，放於 <i>Type</i> 中
ABConvert(<i>Angle</i>, <i>Binary</i>)	將輸入的角度值，以 360 度對應到 256 的刻度值，由 <i>Binary</i> 輸出， <i>Angle</i> 可以輸入-32768~32767 間的任意整數，回傳的 <i>Binary</i> 為-23302~23301 間的整數
BAConvert(<i>Binary</i>, <i>Angle</i>)	將輸入的 <i>Binary</i> ，以 256 對應到 360 的比例，轉為角度直存放在 <i>Angle</i> 中， <i>Binary</i> 可以輸入-23302~23301 間的任意整數，回傳的 <i>Angle</i> 為-32768~32767 間的整數
SaveCalVal(<i>Mode</i>, <i>X0G</i>, <i>X1G</i>, <i>X-1G</i>, <i>Y0G</i>, <i>Y1G</i>, <i>Y-1G</i>, <i>Z0G</i>, <i>Z1G</i>, <i>Z-1G</i>)	根據 <i>Mode</i> 設定各模式的校正值， <i>Mode</i> 請輸入 0~3 間的整數值， <i>X0G</i> ， <i>X1G</i> ， <i>X-1G</i> 請分別輸入在 X 軸向感測到 0g，1g 與-1g 時以 GetXADVal 取得的電壓值， <i>Y0G</i> ， <i>Y1G</i> ， <i>Y-1G</i> 請分別輸入在 Y 軸向感測到 0g，1g 與-1g 時以 GetYADVal 取得的電壓值， <i>Z0G</i> ， <i>Z1G</i> ， <i>Z-1G</i> 請分別輸入在 Z 軸向感測到 0g，1g 與-1g 時以 GetZADVal 取得的電壓值， <i>Mode</i> 以外各輸入值請輸入 0~65535 間的整數 ※
LoadCalVal(<i>Mode</i>, <i>X0G</i>, <i>X1G</i>, <i>X-1G</i>, <i>Y0G</i>, <i>Y1G</i>, <i>Y-1G</i>, <i>Z0G</i>, <i>Z1G</i>, <i>Z-1G</i>)	根據 <i>Mode</i> 取得各模式的校正值， <i>Mode</i> 請輸入 0~3 間的整數值， <i>X0G</i> ， <i>X1G</i> ， <i>X-1G</i> 將回傳 X 軸 0g，1g 與-1g 時的電壓校正值， <i>Y0G</i> ， <i>Y1G</i> ， <i>Y-1G</i> 將回傳 Y 軸 0g，1g 與-1g 時的電壓校正值， <i>Z0G</i> ， <i>Z1G</i> ， <i>Z-1G</i> 將回傳 Z 軸 0g，1g 與-1g 時的電壓校正值，各回傳值將回傳 0~65535 間的整數
RestoreCalVal()	使用預設的校正值替代現在的校正值

※設定校正值時，請先根據圖 4，量測各軸向於三個狀態(0g，1g，-1g)下的數位電壓值，由於電壓值會跳動，建議取十點以上，並去除最大最小值做平均，量測時請固定好模組，以免晃動造成數值錯誤。

模組提供應用事件：

事件名稱 (Event)	啟動條件
XForceLimitEvent	在執行 EnableXForceLimitEvent ()後，當偵測到 X 軸向受力超過設定值時，就會啟動。
YForceLimitEvent	在執行 EnableYForceLimitEvent ()後，當偵測到 Y 軸向受力超過設定值時，就會啟動。
ZForceLimitEvent	在執行 EnableZForceLimitEvent ()後，當偵測到 Z 軸向受力超過設定值時，就會啟動。
ForceLimitEvent2D	在執行 EnableForceLimit2DEvent ()後，當偵測到 2D 平面受力超過設定值時，就會啟動
ForceLimitEvent3D	在執行 EnableForceLimit3DEvent ()後，當偵測到 3D 受力超過設定值時，就會啟動
DevAngleLimitEvent2D	在執行 EnableDevAngleLimitEvent2D ()後，每當偵測到的方

	位，與設定的基準方位，差異大於 SetDevAngleLimit2D() 指令所設定的角度時，就會啟動。基準方位可以用 SetDevAngleNum2D() 指令設定。
ForceRefershEvent	在執行 EnableRefreshEvent () 後，當量測值更新後，就會啟動

範例程式:

```

Peripheral myG As Accelerometer3A @ 0          ' 設定模組編號為 0

Dim g_bStatus As Byte                        ' 儲存狀態回傳值
Dim g_wADx, g_wADy, g_wADz As Word           ' 儲存 X, Y, Z 軸的電壓值
Dim g_iFx, g_iFy, g_iFz As Integer           ' 儲存 X, Y, Z 軸的力值
Dim g_wF2D As Word                           ' 儲存 2D 平面的力值
Dim g_wAngle As Word                         ' 儲存 2D 力的角度
Dim g_wF3D As Word                           ' 儲存 3D 的力值
Dim g_wAngle1 As Word                        ' 儲存 3D 的力值在 XY 平面的夾角
Dim g_bAngle2 As Byte                        ' 儲存 3D 的力值與 Z 軸的夾角
Dim g_bLimit As Byte                         ' 儲存事件偵測值

Sub Main()
    Dim i As Byte                            ' 儲存迴圈計數參數

    Pause 1000

    myG.SetMode(0)                            ' 設定靈敏度模式為 0，適合量測正負 1.5g 的加速度值
    myG.SetRefreshFreq(3)                     ' 設定更新速率為每秒更新十次
    myG.SetAxis2D(0)                          ' 設定 2D 平面為 XY 平面
    Debug CLS
    Pause 1000

    ' For 迴圈取得十次各軸電壓值
    For i=1 To 10
        ' DoLoop 迴圈確認取得更新值
        Do
            g_bStatus = myG.GetRefreshStatus() ' 取得更新狀態
        Loop Until g_bStatus=1

        myG.GetXADVal(g_wADx)                  ' 取得 X 軸加速度的電壓值
        myG.GetYADVal(g_wADy)                  ' 取得 Y 軸加速度的電壓值
        myG.GetZADVal(g_wADz)                  ' 取得 Z 軸加速度的電壓值
        Debug "X: ", g_wADx, ", Y: ", g_wADy, ", Z: ", g_wADz, CR
    Next

    myG.GetXYZForce(g_iFx, g_iFy, g_iFz)       ' 取得 XYZ 三軸的加速度值

```

```

myG.GetForce2D(g_wF2D, g_wAngle)          '      取得 XY 平面的力值與角度

myG.GetForce3D(g_wF3D, g_wAngle1, g_bAngle2)  '      取得 3D 的力值與角度


Debug "Force X: ", g_iFx, ", Y: ", g_iFy, ", Z: ", g_iFz, CR
Debug "2D Force: ", g_wF2D, ", Angle: ", g_wAngle, CR
Debug "3D Force: ", g_wF3D, ", Angle1: ", g_wAngle1, ", Angle2: ", g_bAngle2, CR


myG.SetForceLimit2D(0, 800)                '      設定 2D 平面受力極限為 800(1g)
g_bLimit = 0
myG.EnableForceLimit2DEvent()              '      啟動 2D 受力極限事件

'      DoLoop 迴圈直到偵測到 2D 受力超過極限(1g)才會跳出
Do
Loop Until g_bLimit=1

Debug "Finish"
End Sub

Event myG.ForceLimitEvent2D()
    myG.GetForce2D(g_wF2D, g_wAngle)        '      取得 XY 平面的力值與角度
    Debug "Event Force: ", g_wF2D, CR
    g_bLimit = 1
End Event

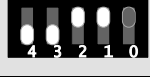
```

附錄

1. 已知問題:

- v1.0: **DevAngleLimitEvent2D** 在設定軸向為 XY 以外的值時，仍只以 XY 軸向作為判斷事件的產生條件。
- v1.1: **GetAngle3D** 取得的 3D 角度，在 $Z > 0$ ，XY 接近 0 時，會誤判為 90 度，當 $XY = 0$ ， $Z < 0$ 會回傳錯誤值。

2. 模組編號開關對應編號表:

	0		8		16		24
	1		9		17		25
	2		10		18		26
	3		11		19		27
	4		12		20		28
	5		13		21		29
	6		14		22		30
	7		15		23		31