**Ameba特異功能集(一)省電大作戰**

作者：MAKERPRO技術顧問 柯大

2016/1/23

資料參考來源：http://www.amebaiot.com

**前言**

為了創造更輕便電的物聯網裝置，瑞昱阿米巴Ameba物聯網加入了更多裝置端的省電模式，並提供Arduino 開發環境不同的省電API，讓裝置可以輕易的以電池或太陽能板供電維持一段長時間的正常運作。

**省電方式一：**

[**Realtek IoT/Arduino Solution**](http://www.amebaiot.com/) **Version 1.0.5 – 2016/02/19**

**新增了對開發板省電 API功能**

**提供了6組API**

void sleep(); //進入sleep

void active(); //關掉sleep()，不允許sleep

void deepsleep(uint32\_t duration\_ms); //進入最省電的模式深睡模式

void setPllReserved(bool reserve); //預設PLL clock關掉，防止其他週邊停止運作void softReset(); //軟體重開機

bool safeLock(); //

要使用Ameba 省電API功能需於程式中加入 library：  
#include <PowerManagement.h>  
API function 說明如下：

static void sleep();

呼叫這個function之後，就允許系統在閒暇的時候進入sleep，（但不一定會馬上進入sleep）

如果Ameba開著wifi，wifi依照802.11的省電模式定時起來並且醒著聽AP發出來的beacon

收完之後如果沒事就會繼續進入sleep

如果wifi資料量比較大的時候，wifi會離開802.11的省電模式，此時系統也不會進入sleep

所以開著wifi的耗電量會因為資料量與wifi環境的乾淨程度有關

static void active();

等於是關掉sleep()，閒暇時不允許sleep

static void deepsleep(uint32\_t duration\_ms);

是最省電的模式，幾乎都關了，醒來之後會從頭開始執行，像是經過reset一樣，呼叫這個function之後會馬上進入deepsleep。

參數duration\_ms：進入deepsleep之後多久之後醒來。

 static void setPllReserved(bool reserve);

當Ameba進入sleep，如果PLL clock關掉，會讓一些週邊(Ex. UART, I2C, SPI…)無法收資料

目前預設是保留PLL 不關掉，適合關掉的情況是，沒有用到這些週邊，或是用到這些週邊但不需要在Ameba進入sleep之後還收資料，關掉省掉的電大約是5.5mA，算是需要自己取捨的電量

static void softReset();

呼叫之後馬上重開機，這是軟體行為，電路上並沒有重新上電

static bool safeLock();

當Ameba進入deepsleep，會無法燒錄程式

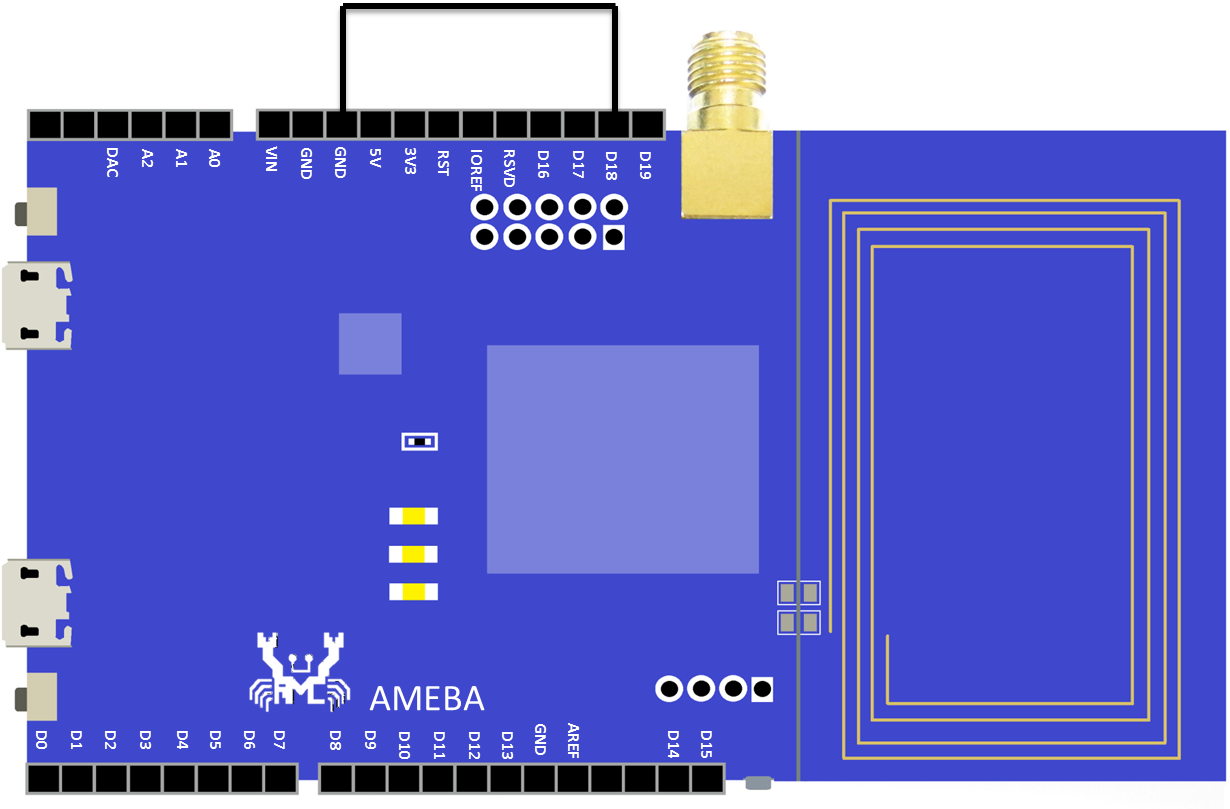
為了避免使用者寫出程式是開啟Ameba就進入deepsleep的程式，所以使用D18這根PIN當做保護，當D18這根PIN被接到GND，就會防止Ameba進入deepsleep或sleep

傳回值 :

true: D18 is connected to GND

         false: D18 is not connected to GND

**※因為Ameba進入deepsleep之後，就無法上傳程式。在deepsleep的API裡面我們將D18這根腳當作避免Ameba進入deepsleep的pin腳。只要將D18接GND，並且重開Ameba 即可重新燒錄新程式。**



**Ameba的省電模式**

Ameba提供許多省電模式，最常用的是deepsleep與sleep。

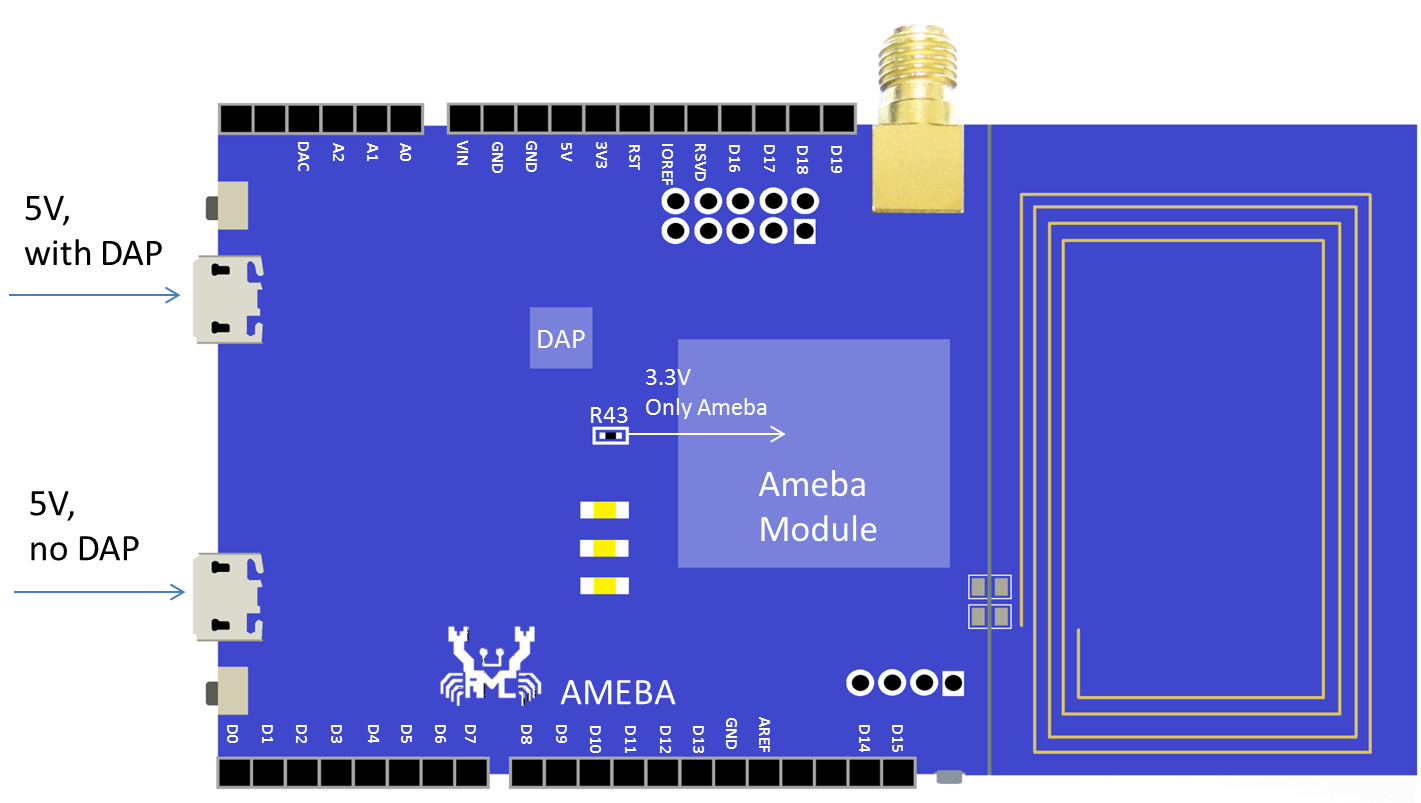
* Deepsleep

當Ameba進入deepsleep，會讓Ameba處於最省電的情況, 它的耗電會小於20 uA (微安培)，此時大部份元件都關掉了，但低解析度的timer仍在運作並且用來喚醒Ameba。這種模式適合定時運作的情境，像是每小時起來量測溫濕度，送出資料，再回到deepsleep。

* Sleep

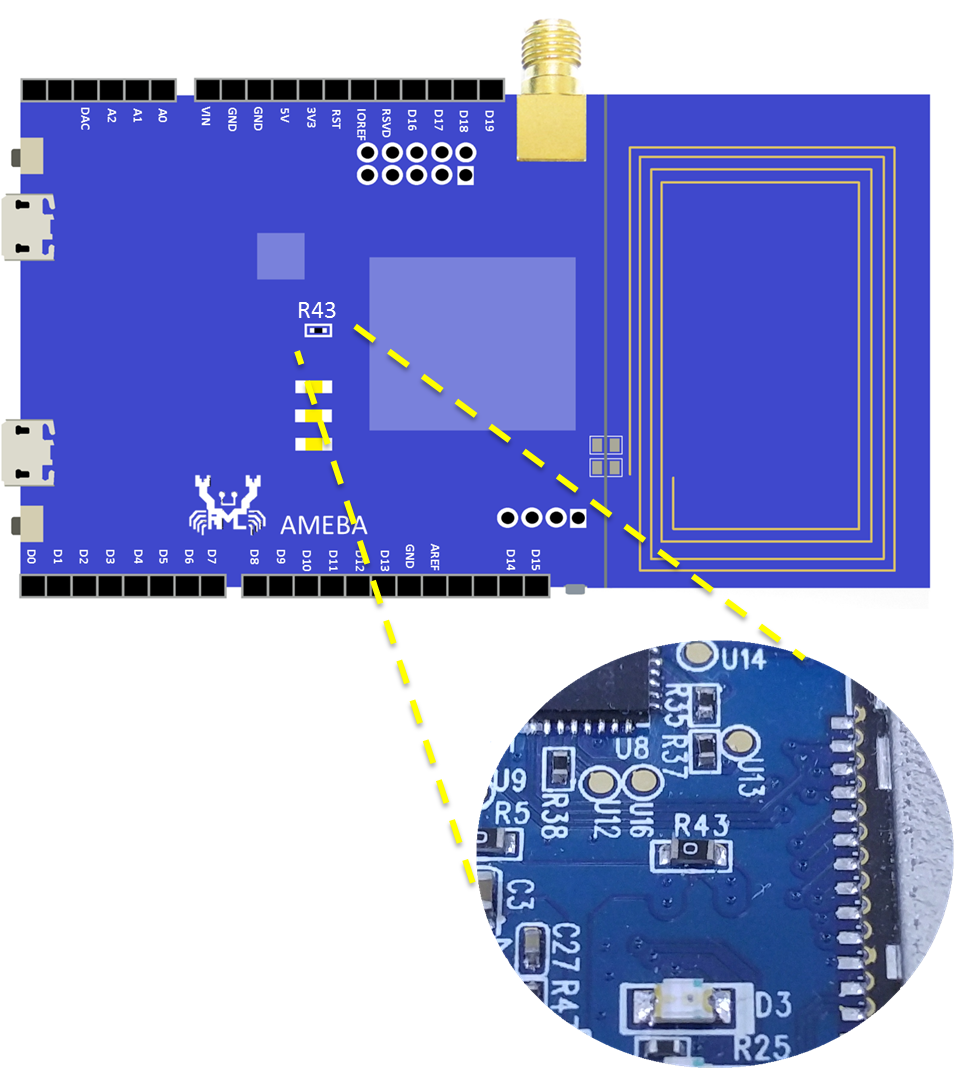
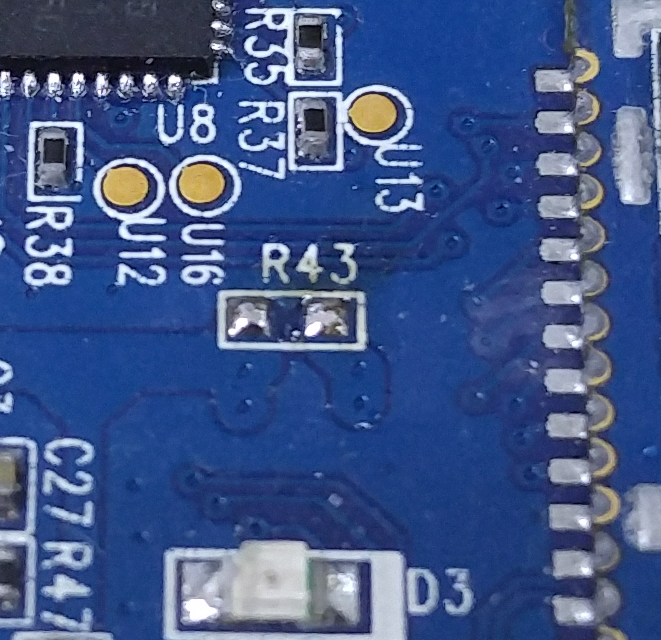
這個模式是Ameba可以維持Wifi連線的省電模式，它的耗電會因為wifi資料量與無線環境的乾淨程度而有所不同，但是瞬間最低電流會在2.5 mA (毫安培)，與wifi起來運作之後的電流平均之後，仍然是相當省電的模式。這種應用適合需要遠端操控Ameba的情境，像是使用手機透過網路通知Ameba馬上量一次溫濕度，再回傳結果。

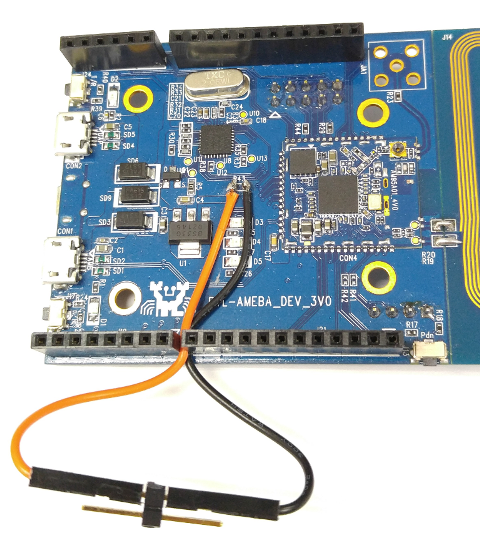
Ameba的供電方式

在Ameba的開發板上, 我們都是使用下圖左上方的USB供電，這裡的USB供電會提供整塊開發板的電源，包括一個DAP的控制元件，LED指示燈，周邊的電容與電阻，以及經過電壓轉換之後，提供3.3V工作電壓給Ameba。  


如果我們單獨供應3.3V的電源給Ameba，而不是使用USB供電給整塊板子，Ameba還是可以運作的。板子上其它元件讓使用者便於開發，當開發完成之後，就可以只供電給Ameba。

如何單獨供電給Ameba

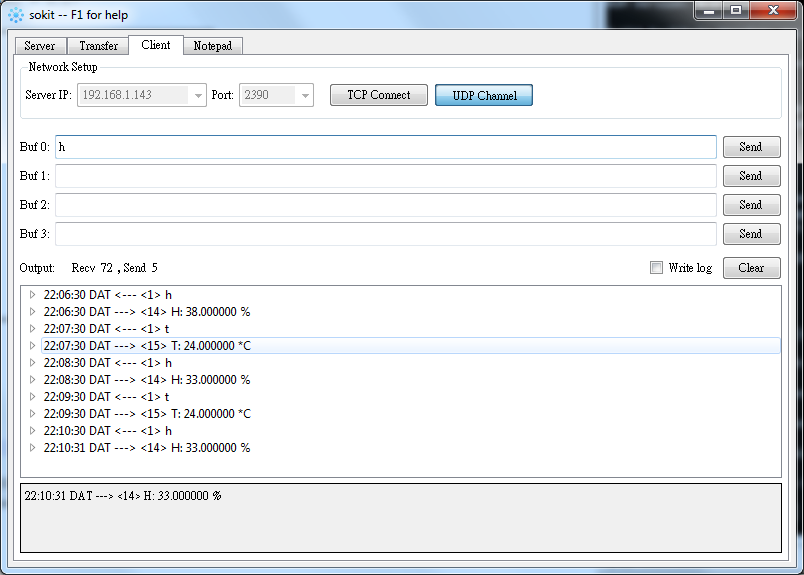
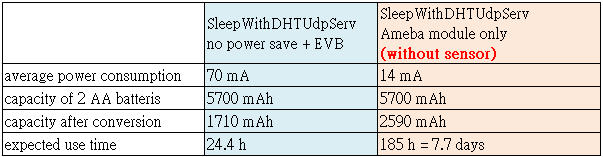
單獨供電給Ameba需要對R43這個電阻做處理，不想更動硬體或不需要體驗到非常省電的可以跳過這一章。  
板子上R43的地方是個電阻，電阻左邊是經過降壓模組的3.3V電源, 這個3.3V電源只單獨供應給電阻右邊的Ameba Module。  
我們需要解焊R43這個電阻，它會露出左右各一個金屬連接點，我們再將左右分別焊上杜邦線即可。  
解焊的方式，需要將R43的左邊與右邊同時受熱，才有辦法將R43這個電阻取下。比較浪費資源但快速的方式是將焊錫塗滿R43，讓焊錫覆蓋到左邊與右邊，這樣就可以將R43取下。  


取下之後，會看到兩個接點。為了之後使用上方便, 我們將左邊與右邊都焊上杜邦線。焊完之後將兩條線做固定，可以黏膠帶、使用熱融膠、或是像下圖卡住固定。  


如圖中，這樣我們一樣可以使用USB供電。當我們想單獨對Ameba供電時，將紅線與黑線分開，由黑線供應3.3V的電源，以及接上電源的GND即可。

**Ameba的省電範例程式**

## ****一、讓Ameba處於Wifi連接的省電模式****

範例說明  
Ameba的deepsleep無法讓wifi處於連線狀態，如果想讓Ameba省電，並且保持wifi連線，可以使用sleep API。  
打開範例 “File” -> “Examples” -> “AmebaPowerSave” -> “SleepWithDHTUdpServ”  
範例裡一樣需要設定連線AP的ssid & password  
這個範例裡，Ameba會先建立wifi連線，進入sleep省電模式，Ameba會當作UDP server，每當接收到UDP封包內容為 ‘H’，就回傳濕度，接收到UDP封包為 ‘T’，就回傳溫度。  
與deepsleep不同的地方是，當Ameba打開sleep模式，會自己找尋可以睡眠的時間，當裝置閒置時就會進入睡眠。整個睡眠與醒來的動作是自動的，不需要使用者干預。  
當Ameba連線之後，我們參考之前測試TCP/UDP的工具Sokit設定Ameba的IP & port。  
我們每分鐘輪流問一次溫濕度，可以看到Ameba都有回應。  
這個範例裡會偵測D18這根防止進入睡眠的pin腳，如果D18被接地，就不會讓Ameba進入sleep睡眠模式，於是我們可以測量兩種耗電：**(實驗所使用的電錶為Keysight 34465A)**  


**NOTE: 實際情況裡，不會用到那麼久，它會因為升壓轉換造成能量損失，也會因為電池電壓隨時間而降低造成無法使用到全部的容量。**  
這裡的數據會因為網路狀況而有不同，但讓Ameba保持連線的情況下會比讓Ameba進入deepsleep要耗電。但讓Ameba保持連線也提供使用者做遠端操控，這是deepsleep無法做到的。這端看使用者的應用而定。

## ****二、將DHT資料上傳到LASS之後進入睡眠****

範例說明  
使用Ameba通常都會使用到它的Wifi功能，實際的省電使用情況也會因為使用wifi功能而有不同。  
這個範例裡，會先量測DHT的溫濕度數據，打開wifi並連上AP，取得NTP時間，連上LASS的MQTT server，將量測的溫濕度數據上傳，再進入deepsleep。每10分鐘重覆這樣的動作。  
打開範例 “File” -> “Examples” -> “AmebaPowerSave” -> “DeepSleepWithDHTLass”  
修改程式裡相關的設定，包括：  
– DHT的型號：DHT11/DHT22/DHT21  
– 連上AP的方式：ssid, password  
– 目前的GPS位置：latitude & longitude  
編譯並上傳之後，量測到的耗電會因為許多因素而影響，包括wifi網路環境是否乾淨、網路連線是否順暢、Server是否馬上回應。但一般應該會在十多秒內完成。  
範例測試的結果裡，我在上傳所花的時間是13s，量得的Ameba module耗電平均為1.3mA。  
這個數據比範例 “DeepSleepWithDHT” 要大上許多，因為Ameba在工作模式下的耗電通常大於29mA，甚至在wifi連線時至少會大於68mA，與deepsleep的0.018 mA相比，我們要讓Ameba儘量待在deepsleep才會取得最大的省電效益。  
為了比較未省電的情況，我們將程式碼修改如下，讓它平常保持網路連線，每10分鐘量一次，量完再上傳至LASS的MQTT Server

void setup()

{

dht.begin();

initializeWiFi();

initializeMQTT();

}

void loop()

{

if (gatherHumidityAndTemperature() == DATA\_CNT\_FOR\_UPLOAD) {

retrieveNtpTime();

if (!client.connected()) {

reconnectMQTT();

}

publishResults();

pDhtData->dataCount = 0;

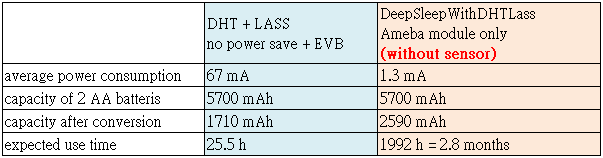
}

// store data back to flash memory

FlashMemory.update();

delay(measureInterval \* 1000);

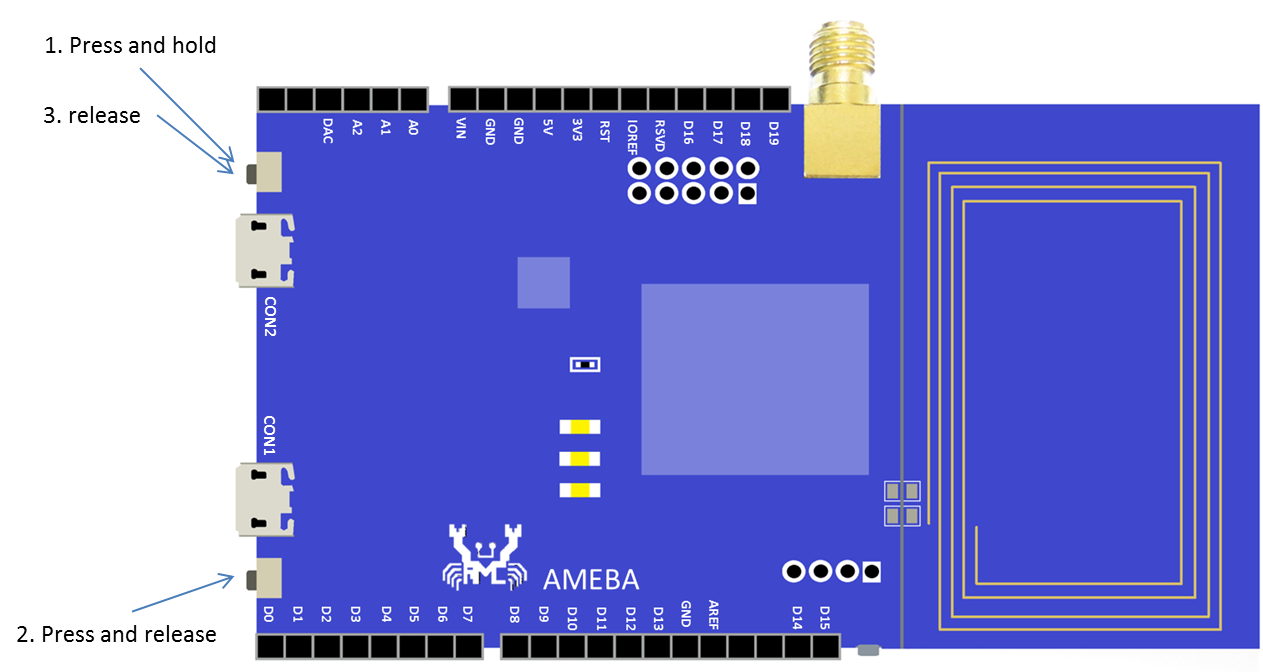
}

這種方式量測到的平均耗電為67mA  
我們用2顆AA電池來比較耗電 **(實驗所使用的電錶為Keysight 34465A)**  
  
可以看到沒省電的情況下只能用一天，但經過省電並且單獨供電給Ameba module可以使用2.8個月。

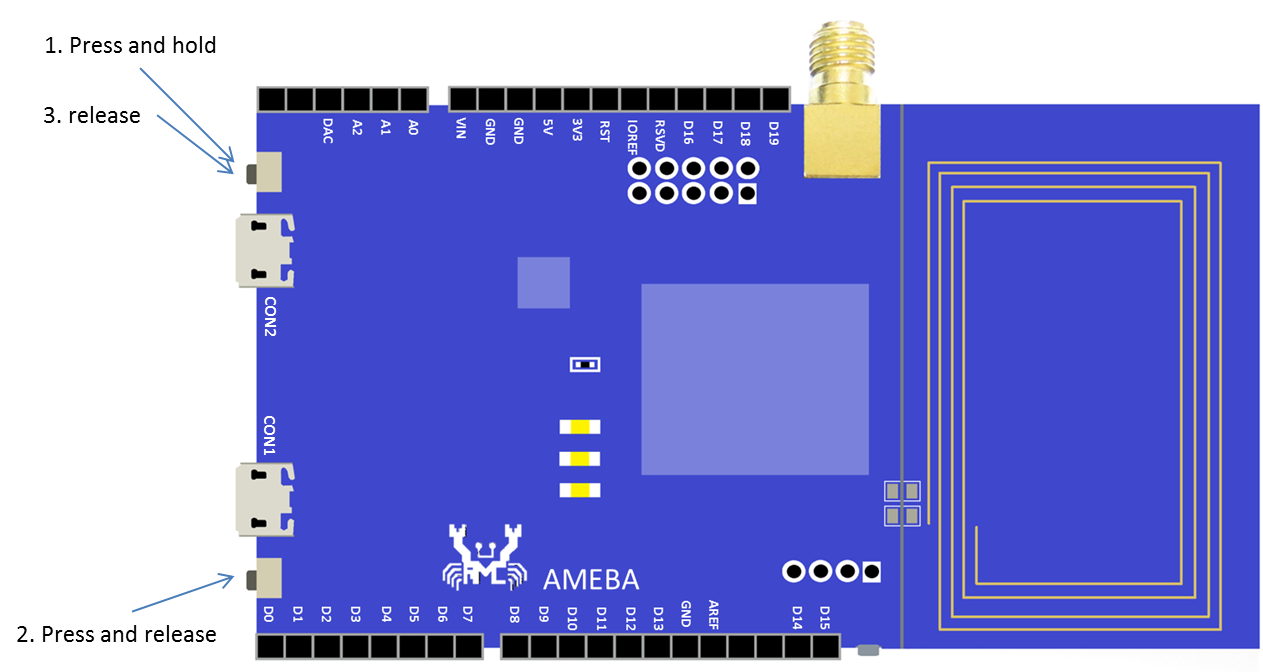
**NOTE: 實際情況裡，不會用到那麼久，它會因為升壓轉換造成能量損失，也會因為電池電壓隨時間而降低造成無法使用到全部的容量。**

**省電方式二：**

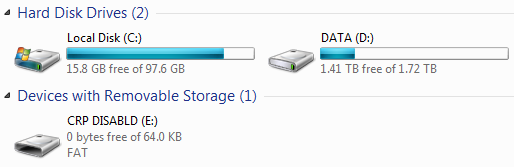
為了更省電，有Ameba技術同好者提出了想將於設計物聯網裝置時將開發板上紅、黃LED閃爍關閉，以達到更省電的目的。  
**針對這個需求可利用下列二個方式將Ameba上的 黃、紅LED 閃爍關閉：  
方式一：.暫時關閉:**  
1.按住CON2旁邊的按鈕不放  
2.按一下CON1旁邊的按鈕再放開



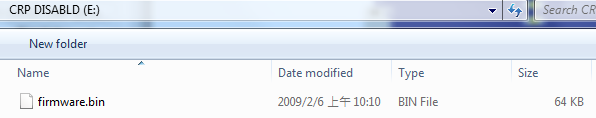
**方式二：.更新DAP Firmware ， LED 永遠關閉(想要更省電的可以用此方式)。**  
更換DAP firmware，請參考以下操作  
1. 按住CON2旁邊的按鈕不放  
2. 按一下CON1旁邊的按鈕再放開  
3. 放開CON2旁邊的按鈕



4.此時會出現一個磁碟槽，上面的標籤為 “CRP DISABLED”



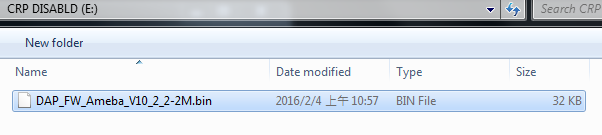
打開這個磁碟，裡面有個檔案 “firmware.bin”，它是目前這片Ameba使用的DAP firmware。



5.要更換firmware，可以先將這個 firmware.bin 備份起來，然後刪掉，再將新的DAP firmware用檔案複製的方式放進去。

6.新的firmware 下載點：  
<https://github.com/Ameba8195/Arduino/raw/master/misc/dap_firmware/>

7.將新的firmware: DAP\_FW\_Ameba\_V10\_2\_2-2M\_LED\_Disable 拷貝到 標籤為 “CRP DISABLED” 的磁碟槽上。



8.最後將USB重新插拔，新的firmware就生效了。  
9.現在開始Ameba板子上的 黃、紅LED 閃爍即永遠關閉了。  
10.可參考按照下列連結：  
<http://www.amebaiot.com/change-dap-firmware/>

**11.更新firmware前最好將原來的firmware.bin備份起來，以便以後可以再恢復可以讓LED閃爍。**

**12.使用 Macbook的Mac上利用VMware建Windows的人，請把檔案在Mac抓下來丟到 Vmware (XP) 裡面，再從Vmware直接將檔案寫到 CRP DISABLD 的資料夾，這樣就會成功了，原本備份的檔案 如果從 Mac端 會產生檔案太大寫不進去狀況，但如果丟到 VM裡面去寫ＯＫ了。**

**預告：Ameba特異功能集(二)忠實看門狗WatchDog Timer( WDT) 待續…..**