Ameba特異功能集(二)忠實看門狗WatchDog Timer( WDT)

作者：MAKERPRO技術顧問 柯大

2016/1/23

資料參考來源：http://www.amebaiot.com

**前言**

物聯網裝置對於感測資料的蒐集常需1年365天，1天24小時不間斷的將感測值透過不同的通訊網路傳輸至雲端。

但因網路狀態不穩或因其他外在的雜訊、電磁干擾，或程式本身長時間運作的問題，或各種情況等，導致裝置系統無法如預期正常運作，變成當機，碰到這種情形，也只能靠人工去對裝置重新開機，或按下重新開機鈕Reset。

在萬物聯網的情形下，您的物聯網裝置可能安裝於一些外部環境，例如架設在高架道路上，用來偵測與累計超過多少高度的車輛每天有多少輛，若這樣的系統當機了，就得派人去現場按重新開機，造成很大的不便及危險性及人力成本。或者在工廠輸送帶的生產線上，設置一個控制或感測生產線運作狀況的裝置，如果這毎裝置當機了，就必須停止整條生產線，等待重新開機，才能恢復原有生產程序，這樣影響太大了。

**解決方案：將物聯網裝置加入看門狗(WatchDog Timer)功能**

阿米巴Ameba物聯網開發板針對Arduino IDE開發環境，在V1.0.5版加入了Watchdog Timer功能，可針對上述這種狀況，讓開發者利用Watchdog timer API功能，自動偵測裝置運作狀況，在程式中加入Watch Dog Timer( WDT）自動計時，遇有系統故障或連網沒回應超過所要求的時間時，就可自動reset 裝置重新開機，也就可以解決上述問題的。

有關對WatchDog Timer的說明可參考下列網站：  
1. 維基百科：<https://zh.wikipedia.org/wiki/>看門狗計時器

2.Arduino.cc網站有關WatchDog 功能

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

**新增了對開發板可設定自動Reset 的Watchdog Timer API功能**

**提供了3組API：**void wdt\_enable(timeout); //開啟看門狗Watchdog計時器功能，並設定重新timeout reset時間

void wdt\_reset(); //清除watchdog計時器的值，重新依前面wdt\_enable(timeout)所設定timeout reset時間倒數計時

void wdt\_disable() //關閉看門狗Watchdog功能

要使用Ameba Watchdog API功能需於程式中加入 library：  
#include "wiring\_watchdog.h"

**範例如下：**

**範例一：**

#include "wiring\_watchdog.h" // 加入Watchdog API library

int timeout = 3500; // 設定Watchdog Timeout 自動Software reset 功能，

最大值是8000ms = 8 秒。

void setup() {

Serial.print("set watchdog timeout: ");

Serial.print(timeout);

Serial.println(" ms");

wdt\_enable(timeout); //開啟watchdog計時器功能，並設定重新timeout reset時間

}

void loop() {

for (int i=1; i<=8; i++) {

Serial.print("delay ");

Serial.print(i);

Serial.println("s");

delay(i\*1000); //當delay 超過所設定timeout 值，Ameba的watchdog 自動對Ameba 板做software reset

Serial.println("feed watchdog");

wdt\_reset(); //清除watchdog計時器的值，重新計算timeout reset時間

}

}

**範例二：**

#include "wiring\_watchdog.h" // 加入WatchDog Timer API library

int timeout = 3500; // 設定Watchdog Timeout 自動Software reset 功能，

最大值是8000ms = 8 秒。

void setup() {

Serial.print("set watchdog timeout: ");

Serial.print(timeout);

Serial.println(" ms");

wdt\_enable(timeout); //開啟watchdog計時器功能，並設定重新timeout reset時間

}

void loop() {

wdt\_reset(); // 進入Ameba執行廻圈，清除watchdog計時器的值，重新計算timeout reset時間

getSystemDatetime();

wdt\_enable(1000); //開啟watchdog計時器功能，並設定重新timeout rese時間=1秒

wdt\_reset(); // 進入一段SD卡存取程式，加入wdt\_reset()清除watchdog計時器的值，重新計算timeout reset時間，如果下面讀取錯誤造成當機，系統即會自動reset

if (SD.exists(logDir)) { // SD卡偵測目錄是否存在

}

else {

SD.mkdir(logDir); // SD卡建立log目錄

}

Serial.println(F("Start state#1."));

wdt\_enable(3000); //開啟watchdog計時器功能，並設定重新timeout rese時間=3秒

wdt\_reset(); // 進入一段讀取電表程式，加入wdt\_reset()清除watchdog計時器的值，重新計算timeout reset時間，如果下面讀取錯誤造成當機，系統即會自動reset

Meter\_no =1 ;

readAPMRdata(); // 讀取電表副程式

trans2value();

displayData();

Serial.println(F("Start state#2."));

wdt\_enable(2000); //開啟watchdog計時器功能，並設定重新timeout rese時間=2秒

wdt\_reset(); // 進入一段寫入記錄程式，加入wdt\_reset()清除watchdog計時器的值，重新計算timeout reset時間，如果下面讀取錯誤造成當機，系統即會自動reset

writelogFile(); // 寫入記錄副程式

Serial.println(F("Start state#3."));

wdt\_enable(2000); //開啟watchdog計時器功能，並設定重新timeout rese時間=2秒

wdt\_reset(); // 進入ftp傳輸程式，加入wdt\_reset()清除watchdog計時器的值，重新計算timeout reset時間，如果下面讀取錯誤造成當機，系統即會自動reset

ftpBackup(); // ftp 上傳資料副程式

wdt\_disable(); //廻圈結束關閉看門狗Watchdog功能

} // End of Loop()

**預告：Ameba特異功**能集**(三) 行動支援的NFC 待續…..**