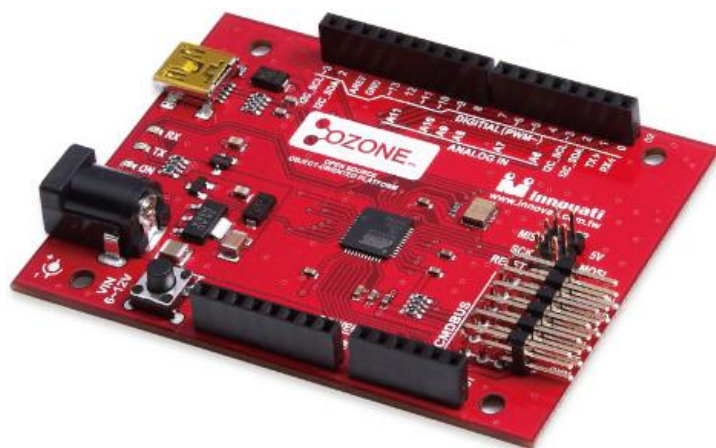


OzoneTM

使用手冊

Revision 1.0

Mar. 23, 2013



內 容

Ozone™	i
商 標.....	2
簡 介.....	3
應 用.....	3
特 色.....	3
規 格.....	4
程 式 庫 安 裝 和 使 用.....	6
模 組 連 接.....	8
附錄--- 範例程式.....	8
Ex. 1 ---使用 2X16A LCD 模組.....	8
Ex. 2 --- 使用 Keypad 模組.....	8
Ex. 3 ---使用 Sonar 模組.....	9
Ex. 4 ---使用 Compass 模組.....	9

商 標

Innovati®， 圖案與BASIC Commander®為利基應用科技股份有限公司之

註冊商標。InnoBASIC™, cmdBUS™及Ozone™為利基應用科技股份有限公司之商標。

©2013利基應用科技股份有限公司 版權所有

基於對產品的持續改善，本公司得不經通知隨時變更本資料或本資料中所提及之產品。未經本公司書面同意或授權，不可重製、散布本產品局部或全部內容。

免 責 聲 明

使用者在使用本產品所做的任何應用，使用者須自行承擔一切風險。公司對於因使用本產品所生之直接、間接或附帶損害，包括且不限設備損失、人身安全健康損失、利潤信譽損失，不負任何責任。本公司產品不可使用於救生或相關儀器設備。未滿14歲兒童須有成人陪同方可使用本產品進行相關實驗。

勘 誤

希望使用者會覺得這是一本生動而且實用的使用者手冊。我們花費很多心力於讓這本手冊更加完整而正確的傳達我們希望使用者了解的訊息，然而難免仍有疏漏之處。為了提供使用者手冊提供最新最詳實的資訊，我們會持續改善增補手冊內容。如在本手冊中發現錯誤之處，歡迎利用電子郵件 service@innovati.com.tw 與我們連絡。如有任何相關資訊更新皆會揭露於網站上，請經常瀏覽我們的網站 <http://www.innovati.com.tw>，以便獲知最新資訊。

簡 介

OZONE™採用國際大廠 ATMEL 所生產之 ATmega32u4 高速微處理控制器，其採用目前最普遍之 AVR 控制核心，具有 20 個數位的輸出/輸入點，其中有 7 個可以做為 PWM 的輸出，另外 12 個可以做為類比輸入。其擁有 32KByte Flash 及 2.5KByte SRAM 程式空間，運作時脈高達 16MHz。

OZONE™ 100%完全相容於由 Arduino™所最新推出之 Leonardo™控制器，也可使用目前擁有最高支援度之開放程式碼 Arduino™程式開發環境，其內建 USB 及程式開發階段簡易硬體除錯介面，使用者只需透過 USB 線與 PC 連接，即可透過 Arduino™即時過程除錯功能及程式碼之下載，此外當程式下載完畢，它也可以脫離 PC 而獨立作業。

OZONE™內建多種硬體加速及擴充支援功能，例如 TIMER、UART、I2C、SPI、EEDATA...等，擁有這些功能不僅可讓 OZONE™可加速相關指令執行，更可在同時間進行多種不同程序處理。利基在此一基礎上進一步開發了 CMDBUS™內建功能相關函式庫，擁有多達 50 種不同功能模組可供使用（同時最多 31 個），讓使用者可以直接享受由 OZONE™所帶來之易於上手及快速開發之便利性。

應 用

- 入門控制器學習套件
- 小型科學計劃用控制器
- 互動藝術

特 色

- 輸入電壓 7V~12V
- 20 個數位輸入輸出腳
- 7 個 PWM (Pulse Width Modulation) 通道
- 12 類比輸入 (解析度 10 bits)
- 32K bytes Flash Memory, 其中 4K Bytes 由 bootloader 使用
- 2.5K Bytes SRAM
- 1K Bytes EEPROM
- 時脈輸入 16 MHz @ 5V
- 尺寸: 57.5 mm x 58.6 mm

規格

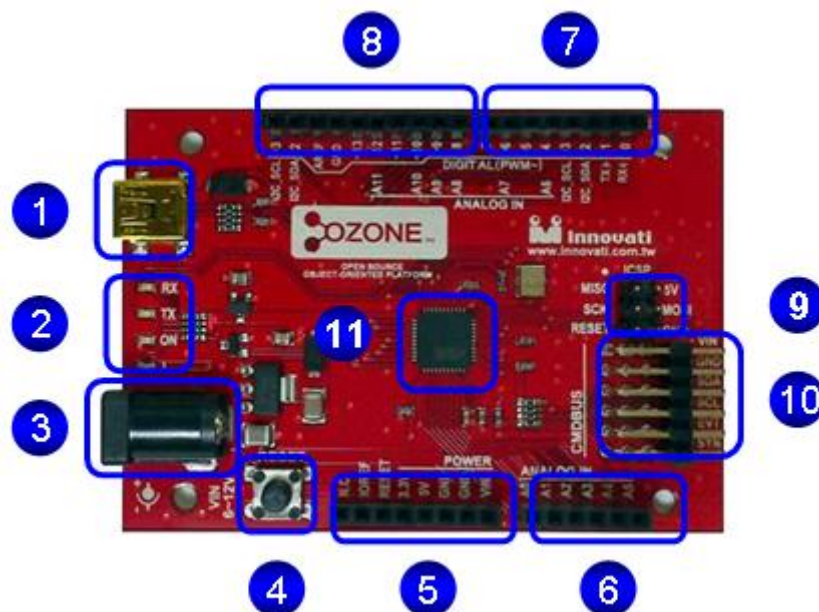


Fig. 1 Ozone™ 控制板

Item	Description
1	Mini USB connector: 可使用 USB 下載程式或除錯
2	4 個 LED 指示燈: RX, TX, ON and L 分別用來指示 UART 接收, 傳送, 開機狀態及 USB 傳輸.
3	DC 6~12V 電源輸入
4	重置鍵. 基本上, 軟體會自動重置 Ozone™ 上的微控制器. 如果無法重置, 可以經由按下重置鍵來啟動 Bootloader. 當按下此鍵時, 微控制器將會被重置且 USB 串列埠將會斷線. 將下來 Bootloader 會重新連線並執行程式. 整個重置過程約需 8 秒.
5	8-pin 連接器母座: N.C.: 未連接 IOREF: I/O 參考電壓 (連接至 5V) RESET: 連接至 RESET button 3.3V: 電路板上的 3.3V 輸出 (300mA) 5V: 電路板上的 5V 輸出 (800mA) GND: 地 Vin: 6~12V 電源輸入
6	6-pin 連接器母座:

	A0: 類比輸入 pin 0 A1: 類比輸入 pin 1 A2: 類比輸入 pin 2 A3: 類比輸入 pin 3 A4: 類比輸入 pin 4 A5: 類比輸入 pin 5
7	8-pin 連接器母座 7: 數位輸入輸出接腳 7 ~6: 數位輸入輸出接腳 6, PWM 輸出及類比輸入 7 ~5: 數位輸入輸出接腳 5, PWM 輸出 4: 數位輸入輸出接腳 4, 及類比輸入 6 ~3: 數位輸入輸出接腳 3, PWM 輸出 2: 數位輸入輸出接腳 pin 2 1 TX: 數位輸入輸出接腳 1, UART 傳輸 0 RX: 數位輸入輸出接腳 0, UART 接收
8	10-pin 連接器母座: ~3: 數位輸入輸出接腳 3 和 I2C SCL 接腳 2: 數位輸入輸出接腳 n 2 和 I2C SDA 接腳 AREF: 類比輸入參考電壓 GND: 地 ~13: 數位輸入輸出接腳 13, 及 PWM 輸出 12: 數位輸入輸出接腳 12 及類比輸入 11 ~11: 數位輸入輸出接腳 11, 及 PWM 輸出 ~10: 數位輸入輸出接腳, PWM 輸出及類比輸入 10 ~9: 數位輸入輸出接腳 9, PWM 輸出及類比輸入 9 8: 數位輸入輸出接腳 8 及類比輸入 8
9	3 個 cmdBUS™ 接腳: 用來連接利基週邊模組, 分別是 VIN, GND, SDA, SCL, EVT and SYN. 連接時請注意腳位對應, 將 Vin 對接 Ozonde 上的 Vin 腳位, 若是腳位錯誤可能造成模組燒毀.
10	ICSP 插座: ICSP, 可使用 SPI 程式庫來做 SPI 傳輸 (Serial peripheral Interface). 插座接腳有 MOSI, MISO, SCK, RESET, GND, 5V.
11	ATmega32U4 微控制器. 16MHz 時脈輸入; 32K bytes FLASH program memory (其中 4K bytes 由 bootloader 使用); 2.5K bytes data SRAM; 1K bytes data EERPOM.

Table 1 Ozon™ 組成

程 式 庫 安 裝 和 使 用

1. 請先安裝 arduino 開發環境

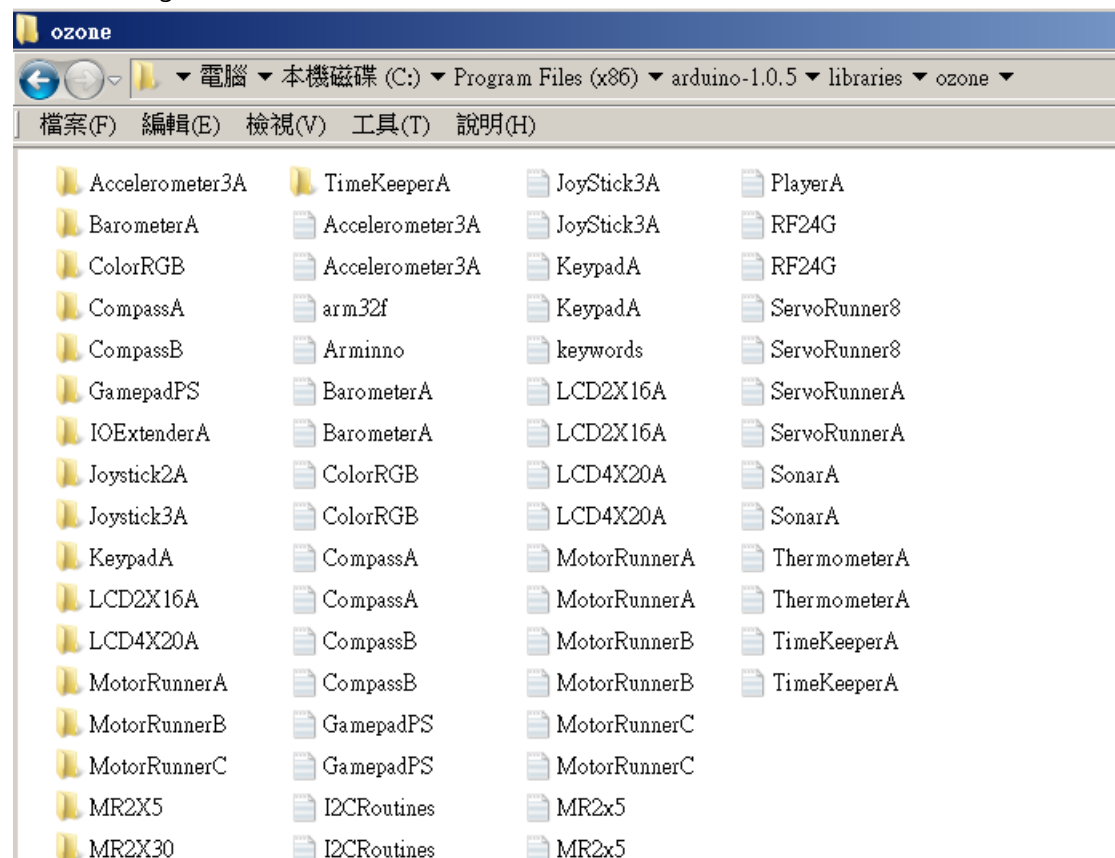
請至 arduino 官網下載(如 Arduino 1.0.5. 以下皆以 1.05 為例)

<http://arduino.cc/en/Main/Software>

2. 安裝 ozone 的專用程式庫

把目錄 ozone 放在 arduino-1.0.5\libraries 目錄中

(例如 C:\Program Files (x86)\arduino-1.0.5\libraries\ozone)



3. 執行 arduino 開發環境

雙擊 arduino.exe



4. 直接使用 Ozone 範例

在功能表中選擇 檔案 – 範例 – ozone

，選擇其中一個範例來執行，就可以使用利基應用科技針對各模組所提供的範例



模 組 連 接

附錄--- 範例程式

以下提供幾個範例程式來協助使用者熟悉 Ozone™ 和週邊模組的連接.

Ex. 1 ---使用 2X16A LCD 模組

```
#include <ozone.h>

LCD2X16A myLCD(0);           // declare Module ID as 0

void setup()
{
    myLCD.BacklightOn(0);      // turn on the LCD backlight
    myLCD.Display("Hello");    // display "Hello"
}

void loop()
{
}
```

Ex. 2 --- 使用 Keypad 模組

```
#include <ozone.h>

KeypadA myKey(1);             // declare Module ID as 1
uint8_t Status, KeyID;

void setup()
{
```

```

myKey.SetKeypadMode(2);      // set keypad to mode 2 (number mode)
}

void loop()
{
  Status = myKey.GetKeyID(KeyID);  // polling keypad input
  if(Status) Serial. print(KeyID);  // display in serial monitor, if key pressed
  delay(200);                      // pause 200ms
}

```

Ex. 3 ---使用 Sonar 模組

```

#include <ozone.h>
SonarA mySonar(16);          // declare Module ID as 16

uint8_t Status;
uint16_t Distance;

void setup()
{
}

void loop()
{
  do {
    mySonar.Ranging() ;        // range finding
    delay(100);                 // pause 100ms
    Status= mySonar.GetDistance(1, Distance);  // get range finding result
  }
  while(Status != 1);           // wait for valid result
  Serial.print("Distance=");   // display text in serial monitor
  Serial.println(Distance);     // display result in serial monitor
}

```

Ex. 4 ---使用 Compass 模組

```
#include <ozone.h>

CompassB myCompass(3);          // declare Module ID as 3

uint16_t wAngle;
void setup()
{
}
void loop()
{
    myCompass.GetAngle(wAngle);
    Serial.print("Angle=");      // display text in serial monitor
    Serial.println(wAngle);      // display result in serial monitor
    delay(200);                  // pause 200ms
}
```