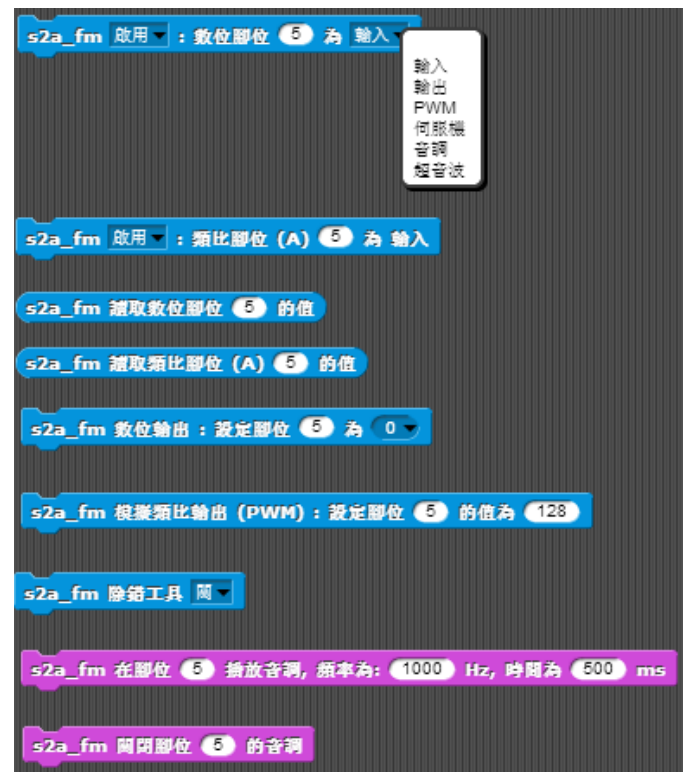
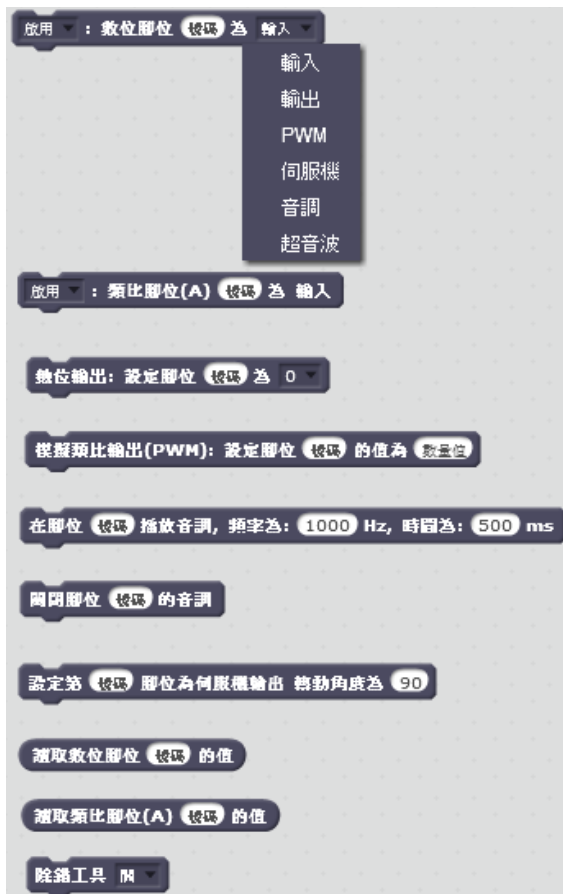


s2a_fm



擴充 Scratch 2.0、Snap! 4.0 的 Arduino 連結功能

Copyright © 2013-15 Alan Yorinks. All rights reserved.

This manual is distributed WITHOUT ANY WARRANTY, without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

July 24, 2015

Added a table in section 8 acknowledging all of the block and documentation translators.

April 12, 2014

Version 1.5 Changes

Korean Translation provided by Dr. Eungil Kim. (August 21, 2014).

Italian Translation provided by Gianfranco Zuliani (Dec 12, 2014).

Greek Translation provided by Alexandros Moskofidis (Feb 2, 2015).

Oct 9, 2014

s2a_fm Version 1.4

Version 1.4 Changes

Taiwanese Translations provided by the courtesy of Hsu Jen-Chieh.

Chinese Translations for Snap! and Scratch have been provided courtesy of Professor

Yu Fang jJun

French Translations for Snap! and Scratch have been provided courtesy of Sebastien Canet

Bug fix for duration field of Sound block. Duration now works properly.

Version 1.3 Changes

1. Support for the Snap!Mobile project has been added.

2. Spanish Translations for Snap! And Scratch have been provided courtesy of Professor José Manuel Ruiz

Version 1.2 Changes:

1. Support for Snap! 4.0 has been added.

2. Support for HC-SR04 type Ping devices has been added.

3. Translation of Scratch Block Text into Dutch has been provided.

4. Facility to translate Scratch/Snap Block Text to any language is included.

Version 1.1 Changes:

1. When enabling a digital pin, the pin is verified to support the requested mode.

2. Added debugging feature

目錄

1.簡介.....	1
2.什麼是 s2a_fm?.....	1
3.安裝 s2a_fm.....	1
3.1.Python 與 Python 相關檔案.....	1
3.2.Arduino 程式.....	2
4.執行 s2a_fm.....	2
4.1.建立 Scratch 專案.....	3
4.2.建立 Snap!專案.....	4
5.擴充積木.....	5
5.1.設定數位腳位模式.....	5
5.1.1.啟用.....	5
5.1.2.停用.....	5
5.1.3.號碼.....	6
5.1.4.數位模式.....	6
5.2.設定類比腳位模式.....	6
5.2.1.啟用.....	6
5.2.2.停用.....	6
5.2.3.號碼.....	7
5.3.數位輸出.....	7
5.3.1.號碼.....	7
5.3.2.輸出值.....	7
5.4.模擬類比輸出(PWM).....	7
5.4.1.號碼.....	7
5.4.2.PWM 值.....	8
5.5.播放音調.....	8
5.5.1.號碼.....	8
5.5.2.頻率 (HZ).....	8
5.5.3.播放時間 (ms).....	8
5.6.關閉音調.....	8
5.6.1.號碼.....	8
5.7.轉動伺服機.....	9
5.7.1.號碼.....	9
5.7.2.角度.....	9
5.8.讀取數位腳位.....	9
5.9.讀取類比腳位.....	9
5.10.超音波.....	9
5.11.偵錯.....	10
5.12.紅色的停止按鈕.....	11
6.記錄檔.....	11
7.範例程式.....	11
7.1.Scratch/Snap! 透過數位腳位 6 開啟 LED.....	11
7.2. Scratch/Snap 透過 A2 腳位上的可變電阻讓 Scratch 小貓旋轉.....	12
8.Scratch/Snap! 德文、西班牙文及中文版積木.....	12
8.1.翻譯成其他語言.....	12
9.簡化程式啟動的 Windows .bat 及 Linux .sh 檔案.....	13

10. 專案目錄結構.....	13
11. 參考資料.....	15
12. 問題、意見及錯誤回報.....	15

1. 簡介

Arduino 玩家！你在找不用寫程式、透過圖形化介面設計 Arduino 的方法嗎？

Scratch 和 Snap! 程式設計師！想連結 Arduino 控制硬體嗎？如果 Scratch 或 Snap! 可以控制實體裝置，像是 LED、車子，或監控溫度、電位器及光線並做出反應，你會想做什麼？

不用再找了，s2a_fm 就是你的答案！

2. 什麼是 s2a_fm？

s2a_fm 是一個 Python 寫的 Scratch 硬體擴充程式，讓 Scratch 可平順地與 Arduino 溝通。

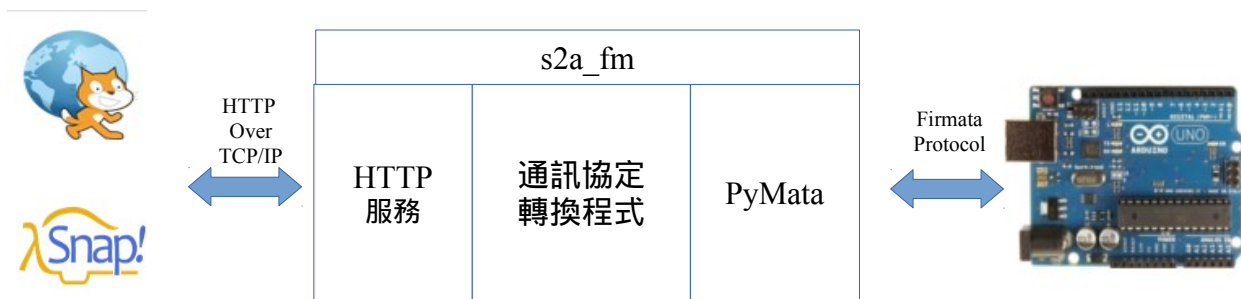


Figure 1 – s2a_fm Architecture

s2a_fm 由三個部分組成：

1. HTTP 服務：負責與 Scratch 溝通。它是一個簡單的 Python BaseHTTPServer。
2. 通訊協定轉換程式：負責 HTTP 與 Firmata 兩種通訊協定的轉換，透過它 Scratch/Snap!/Arduino 彼此可以平順地整合在一起。
3. PyMata：Python 函式庫負責與 Arduino 透過 Arduino 的 Firmata 通訊協定溝通，確保快速、效率、一致。PyMata 會處理所有 Firmata 通訊的細節。

在 Scratch 中請開啟 s2a_fm 套件提供的 s2a_fm_base.sb2 專案，即可在「更多積木」中看到擴充的 Arduino 相關積木。在 Snap! 中請匯入 s2a_fm_Snap_base.xml，即可在「動作」、「聲音」、「偵測」中看到 Arduino 相關積木。

3. 安裝 s2a_fm

3.1. Python 與 Python 相關檔案

在安裝 s2a_fm 之前，請先安裝以下 Python 程式：

- Python 2.7 或以上，可自 Python 官網(<http://python.org/>)下載

- PySerial (<http://pyserial.sourceforge.net/>)
- PyMata (<https://github.com/MrYsLab/PyMata>)

PySerial 及 PyMata 可透過 pip (<http://www.pip-installer.org/en/latest/>) 安裝，若電腦中已安裝 pip 可在「命令列提示字元」下輸入：pip install 套件名稱。

或是透過以上的連結下載套件，接著輸入以下指令安裝 PySerial 及 PyMata：

```
python setup.py install
```

注意：可能需要管理者權限才能安裝。

3.2. Arduino 程式

開啟 Arduino 開發工具，載入「檔案」-「範例」-「Firmata」-「Standard Firmata」。

如果想使用 s2a_fm 提供的音調或超音波 (ping) 功能，請載入 PyMata 套件中的 FirmataPlus 程式。請確認使用的是 PyMata 1.54 或以上的版本。

請將 NewPing 函式庫加入 Arduino 才能成功編譯這段程式。NewPing 函式庫可在此下載：
https://code.google.com/p/arduino-new-ping/downloads/detail?name=NewPing_v1.5.zip。

注意：NewPing 安裝後，請參考以下網頁的指示修改一個檔案：

https://code.google.com/p/arduino-new-ping/wiki/HELP_Error_Vector_7_When_Compiling

NotSoStandardFirmata 已被 FirmataPlus 取代，但還是會提供給有需要它的人。

Arduino 開發環境建議使用 1.5.5 以上版本，較舊本版可能會編譯失敗。

4. 執行 s2a_fm

將 Arduino 透過 USB 埠與電腦連接。

在「命令列提示字元」下切換到 s2a_fm 安裝目錄，輸入：

```
python s2a_fm.py COM_PORT_ID
```

COM_PORT_ID 為 Arduino 使用的埠號，它應該和 Arduino 開發環境中連接的埠號相同。以 Windows 為例可能是 COM3，在 linux 裡可能是 /dev/ttyACM0。

程式執行後應會看到以下訊息：

```
s2a_fm version 1.2    Copyright(C) 2013-2014 Alan Yorinks    All Rights Reserved
```

```
Opening Arduino Serial port /dev/ttyACM0
```

```
Please wait while Arduino is being detected. This can take up to 30 seconds ...
```

```
Board initialized in 1 seconds
```

```
Total Number of Pins Detected = 20
```

```
Total Number of Analog Pins Detected = 6
```

```
Please wait for Total Arduino Pin Discovery to complete. This can take up to 30 additional seconds.
```

Arduino Total Pin Discovery completed in 1 seconds
Starting HTTP Server!
Use <Ctrl-C> to exit the extension

Please start Scratch or Snap!

4.1. 建立 **Scratch** 專案

開啟 Scratch 2.0 點選「檔案」-「開啟」，自本套件的 ScratchFiles/ScratchProjects 目錄下選取 s2a_fm_base_zh_tw.sb2。



偵測到 Scratch 時會在命令提示字元視窗中出現如下訊息：

Scratch detected! Ready to rock and roll...

點選「更多積木」後，將可看到「s2a_fm – Scratch to Arduino」旁出現綠色的圓點表示成功建立連結。



4.2. 建立 *Snap!* 專案

開啟 Snap! 後，點選「檔案」-「導入」。

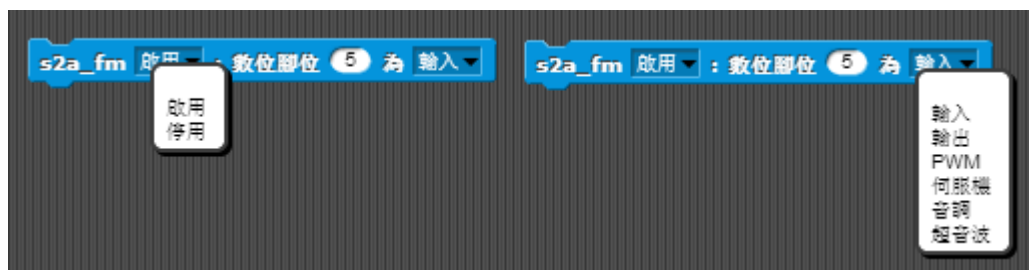


選取本套件「Snap!Files」目錄下的「s2a_fm_Snap_base_zh_tw.xml」，這個空白專案會載入所有 s2a_fm 積木。

5. 擴充積木

以黑白顯示的是 Scratch 積木，以彩色顯示的是 Snap!積木。Snap!中所有 s2a_fm 積木都以 s2a_fm 開頭以示區別。

5.1. 設定數位腳位模式



在 Snap!中，這個積木放在積木面板的「偵測」頁籤。

這個積木可啟用或停用 Arduino 的數位腳位做為輸入、輸出、PWM、伺服機、音調或超音波功

能。

5.1.1. 啟用

在存取腳位的輸入或輸出值前必須先啟用它。只要自下拉式選單中選取「啟用」即可啟用。在選取「輸入」模式時，Arduino 板子會回報腳位讀取到的變化值。

5.1.2. 停用

若腳位之前已被啟用，選取停用將導致此腳位停止回報。一般來說不會如此設定，這個功能提供 Scratch 程式撰寫上的彈性。

5.1.3. 號碼

輸入值必須在 Arduino 板子的腳號範圍內。本套件程式啟動時，在命令提示字元視窗中會顯示偵測到的 Arduino 腳位數，同時也會記錄在記錄檔中(參考 6.記錄檔)。若腳位號碼超出範圍，命令提示字元視窗中會顯示錯誤，同時也會記錄在記錄檔中。偵測到錯誤時，將忽略此命令。

5.1.4. 數位模式

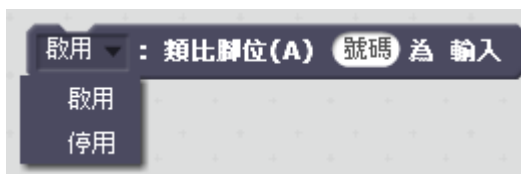
支援六種數位模式：

- **輸入** 連結到輸入裝置，如開關。
- **輸出** 連結到輸出裝置，如 LED。
- **PWM** 用 AnalogWrite 方式輸出。可用來控制 LED 的亮度。
- **伺服機** 用於控制馬達。
- **音調** 透過 Firmata 與 音調函式庫控制腳位輸出。
- **超音波** 使用 NewPing 函式庫。最多可同時監控六個裝置。

注意： HC-SR04 裝置運作模式為單一腳位(trigger 與 echo 連接在一起)，詳情請參考 NewPing 說明文件：https://code.google.com/p/arduino-new-ping/wiki/NewPing_Single_Pin_Sketch。

特別注意： 某些 Arduino 板子提供內部的上拉電阻。若要啟用某腳位的上拉模式請參考 Arduino 說明文件。

5.2. 設定類比腳位模式



類比腳位只能做為輸入，只有啟用或停用兩種功能。

在 Snap! 中，這個積木放在積木面板的「偵測」頁籤。

5.2.1. 啟用

在讀取類比腳位前必須先啟用。自下拉式選單中選取「啟用」，Arduino 將自動回報此腳位的變化值。

5.2.2. 停用

若腳位之前為啟用，選取停用將不再回報變化值。

5.2.3. 號碼

輸入值必須在板子類比腳號的範圍內。本套件程式啟動時，在命令提示字元視窗中會顯示偵測到的 Arduino 腳位數，同時也會記錄在記錄檔中(參考 6.記錄檔)。若腳位號碼超出範圍，命令提示字元視窗中會顯示錯誤，同時也會記錄在記錄檔中。偵測到錯誤時，將忽略此命令。腳號即為 Arduino 類比腳位的代號，如，若要啟用類比腳位 A3，則在積木中輸入 3。

5.3. 數位輸出



在 Snap! 中，這個積木放在積木面板的「偵測」頁籤。

5.3.1. 號碼

此腳位必須先啟用為「輸出」，且腳位號碼必須在板子數位腳位的範圍內。若腳位沒有啟用為輸出，或是超出數位腳號的範圍，將忽略此命令，同時也會在命令提示字元視窗及記錄檔中記錄錯誤。

5.3.2. 輸出值

若數位腳位啟用並設定為輸出，可將它的輸出值設為 1 或 0。可自下拉式選單選取。

5.4. 模擬類比輸出(PWM)



在 Snap! 中，這個積木放在積木面板的「偵測」頁籤。

5.4.1. 號碼

此腳位必須先啟用為 PWM，而且必須是一個 PWM 數位腳號。若腳位沒有啟用為 PWM，或是超出數位腳號的範圍，將忽略此命令，同時也會在命令提示字元視窗及記錄檔中記錄錯誤。

5.4.2. PWM 值

可將它的輸出值設在 0-255 之間，若超出範圍將忽略此命令，同時也會在命令提示字元視窗及記錄檔中記錄錯誤。

5.5. 播放音調



這個積木可讓 Arduino 在某個腳位播放一段音調。

對 Snap! 來說，這個積木放在積木面板的「聲音」頁籤。

5.5.1. 號碼

此腳位必須先啟用為「音調」，而且必須設在數位腳號的範圍內。若腳位沒有啟用為音調，或是超出數位腳號的範圍，將忽略此命令，同時也會在命令提示字元視窗及記錄檔中記錄錯誤。

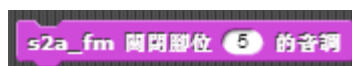
5.5.2. 頻率 (HZ)

播放頻率，不會檢查輸入值。

5.5.3. 播放時間 (ms)

播放時間，以毫秒為單位。若設為 0 將一直重複播放，可使用另一個「關閉音調」積木停止播放。

5.6. 關閉音調



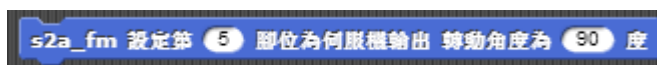
這個積木會停止播放音調。主要用在「播放音調」積木值設為 0、重複播放時，可透過此積木停止播放。

在 Snap! 中，這個積木放在積木面板的「聲音」頁籤。

5.6.1. 號碼

此腳位必須先啟用為音調，而且必須設在數位腳號的範圍內。若腳位沒有啟用為音調，或是超出數位腳號的範圍，將忽略此命令，同時也會在命令提示字元視窗及記錄檔中記錄錯誤。

5.7. 轉動伺服機



這個積木可讓將伺服機轉動指定值。

對 Snap! 來說，這個積木放在積木面板的「動作」頁籤。

5.7.1. 號碼

此腳位必須先啟用為伺服機，而且必須設在數位腳號的範圍內。若腳位沒有啟用為伺服機，或是超出數位腳號的範圍，將忽略此命令，同時也會在命令提示字元視窗及記錄檔中記錄錯誤。

5.7.2. 角度

設定伺服機以角度表示，範圍為 0 到 180，設定值超出範圍將忽略此命令，同時也會在命令提示字元視窗及記錄檔中記錄錯誤。

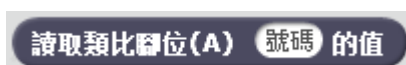
5.8. 讀取數位腳位



這是一個回報數位輸入值的積木，可用來讀取指定腳位的值，回傳 1 或 0。

在 Snap! 中，這個積木放在積木面板的「偵測」頁籤。

5.9. 讀取類比腳位



這是一個回報類比輸入值的積木，可用來讀取指定腳位的值，回傳 0-1023。

在 Snap! 中，這個積木放在積木面板的「偵測」頁籤。腳號即為 Arduino 類比腳位的代號，若要啟用類比腳位 A3，則在積木中輸入 3。

在 Snap! 中，這個積木放在積木面板的「偵測」頁籤。

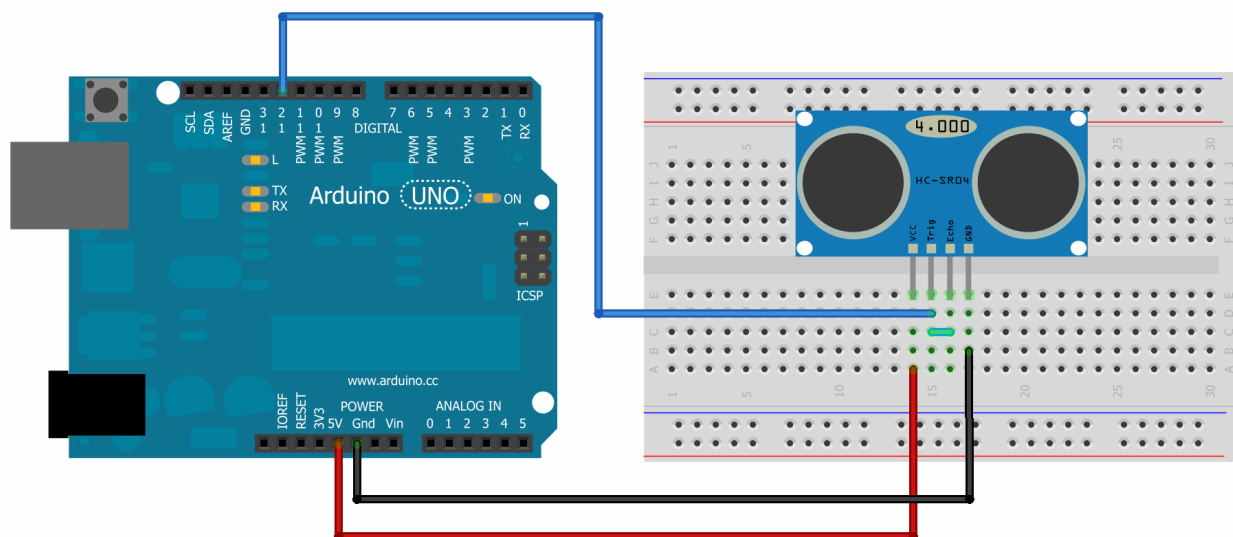
5.10. 超音波

超音波(HC-SR04 超音波感測器)可以公分為單位 監控物件的距離。這個裝置使用一個腳位設定，接線方式如後，腳位先以「設定類比腳位模式」積木設定，再透過「讀取數位腳位」積木讀取監控值。

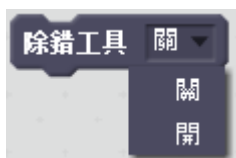
關於 Arduino 的 NewPing 函式庫，請參考「3.2 Arduino 程式」。

啟用超音波對應的腳位後，可透過「讀取數位腳位」積木讀取該腳位的值。

基本的接線方式如下：



5.11. 偵錯



自下拉式選單選取 On 或 Off (預設為 Off) 開啟或關閉偵錯功能。啟用偵錯時，所有 Scratch 指令會同時顯示在命令提示字元視窗及記錄檔中，每個指令含本地時間的時間戳，不含回報積木(偵錯裝置的資訊)的內容，因為 Scratch 每秒約做 30 次讀取，若加入這些資訊將會淹沒記錄檔。

在 Snap! 中，這個積木放在積木面板的「偵測」頁籤。

偵錯記錄檔可能內容如下：

```
DEBUG: 2013-12-19 14:19:06.733833: debugger On
DEBUG: 2013-12-19 14:19:06.768232: digital_pin_mode Enable 48 Output
DEBUG: 2013-12-19 14:19:06.800449: digital_pin_mode Enable 51 Output
DEBUG: 2013-12-19 14:19:06.849955: digital_write 48 1
DEBUG: 2013-12-19 14:19:06.872217: digital_write 51 1
DEBUG: 2013-12-19 14:19:06.906062: digital_write 48 0
```

5.12. 紅色的停止按鈕

點選 Scratch 播放器中的紅色停止按鈕，會傳送一個重設命令給 Arduino，並同時將所有內部資料重設為初始值。

6. 記錄檔

每次 s2a_fm 啟動都會建立一個新的記錄檔，之前的記錄檔將被捨棄。記錄檔的名稱為「s2a_fm_debugging.log」，位於「log」目錄下(可參考「10. 專案目錄結構」)。記錄檔中的「info」為一般操作記錄，其中的偵錯訊息可協助偵錯 Scratch 存取 Arduino 的程式。

如下為記錄檔可能內容：

```
INFO:root:s2a_fm version 1.2 Copyright(C) 2013-2014 Alan Yorinks All Rights Reserved
INFO:root:com port = /dev/ttyACM0
INFO:root:20 Total Pins and 6 Analog Pins Found
INFO:root:Scratch detected! Ready to rock and roll...
DEBUG:root:digital_pin_mode: The pin number must be set to a numerical value
DEBUG:root:analog_write: The value field must be set to a numerical value
```

在這個例子中，INFO 記錄一般操作並提供相關資訊，DEBUG 記錄表示發生問題。在第一個 DEBUG 中，表示腳位在設為數位模式時，腳號設定錯誤。第二個 DEBUG 指出某個類比輸出積木的值設定錯誤。

7. 範例程式

可在 ScratchFiles/ScratchProjects 或 Snap!Files/Snap!Projects 目錄下找到這些範例程式。

注意：以下畫面取自 Scratch 積木。

7.1. Scratch/Snap! 透過數位腳位 6 開啟 LED



7.2. Scratch/Snap 透過 A2 腳位上的可變電阻讓 Scratch 小貓旋轉



8. Scratch/Snap! 德文、西班牙文及中文版積木

感謝我們的使用者將積木翻譯成各種語言。

Sjoerd Dirk Meijer 為 s2a_fm 設計了統一的翻譯機制。

Scratch 的翻譯檔位於 s2a_fm/ScratchFiles/ExtensionDescriptors 與 s2a_fm/ScratchFiles/ScratchProjects 目錄。

Snap!則位於 s2a_fm/Snap!Files 目錄。

感謝以下人士提供語言翻譯		
語言	積木翻譯	文件翻譯
簡體中文	Professor Yu Fang Jun	Professor Yu Fang Jun
正體中文 (TW)	Hsu Jen-Chieh	鄭淑芬(Cheng Shu-Feng)
荷蘭文	Sjoerd Dirk Meije	
法文	Sebastien Canet	
德文	Aldo von Wangenheim	
希臘文	Alexandros Moskofidis	
義大利文	Gianfranco Zuliani	
韓文	Dr. Eungil Kim	
葡萄牙文	Aldo von Wangenheim	
西班牙文	Professor José Manuel Ruiz	Professor José Manuel Ruiz

8.1. 翻譯成其他語言

若要提供其他語言翻譯，有兩個檔案需要修改。第一個是擴充描述檔，另一個是 xlate.cfg。

比較英文版及語言版的檔案即可知道要修改那些字串，接著再將語言資訊加入 xlate.cfg 檔，在每個關鍵字之後加入額外的語言翻譯。注意逗號後不能空白。

例如：

```
ln_INPUT = Input,ingang,Entrada,輸入,MY_NEW_TRANSLATION
```

在以上的例子中，包含了英文、德文、西班牙文、正體中文及一個新的翻譯字串：

MY_NEW_TRANSLATION。

這兩個檔案修改後，在 Scratch 中按著 Shift 鍵選取「檔案」-「匯入實驗性擴充功能」，載入擴充描述檔(這個功能只有離線版才提供)。若檔案修改正確，將可在「更多積木」中看到翻譯後的積木。

若都沒反應，請用JSON lint(<http://jsonlint.com/>)檢查修改內容，進行必要的修正。

9. 簡化程式啟動的 Windows .bat 及 Linux .sh 檔案

在「extra goodies」下有兩個目錄分別提供 Windows 的.bat(s2a_fm.bat)和 linux 的.sh(s2a_fm.sh)，讓啟動 s2a_fm 變簡單點。

10. 專案目錄結構

```
s2a_fm
├── ArduinoFiles
│   ├── README.TXT
│   └── READM.TXT~
├── documentation
│   ├── Español
│   │   └── s2a_fm_Espanish_tutorial.pdf
│   ├── LED_EXAMPLE.png
│   ├── pot1.png
│   ├── s2a_fm_reference.pdf
│   ├── scratch_blocks.png
│   └── snap_blocks.png
├── extra_goodies
│   ├── linux
│   │   └── s2a_fm.sh
│   └── windows
│       └── s2a_fm.bat
├── log
│   └── s2a_fm_debugging.log
├── ScratchFiles
│   └── ExtensionDescriptors
│       ├── s2a_fm_Es.s2e
│       ├── s2a_fm_Fr.s2e
│       ├── s2a_fm_NL.s2e
│       └── s2a_fm.s2e
```

- s2a_fm_zh_cn.s2e
- s2a_fm_zh_tw.s2e
- ScratchProjects
 - s2a_fm_base_Es.sb2
 - s2a_fm_base_Fr.sb2
 - s2a_fm_base_nl.sb2
 - s2a_fm_base.sb2
 - s2a_fm_base_zh_cn.sb2
 - s2a_fm_base_zh_tw.sb2
 - sonarTest.sb2
 - spinning_cat.sb2
 - Turn On LED On Pin 6.sb2

— Snap!Files

- Español
 - blink_tiempo_variable.xml
 - blink.xml
 - boton_con_imagen.xml
 - boton.xml
 - contador1.xml
 - contador2.xml
 - s2a_fm_Snap_base_Es.xml
 - semaforo_tiempo_variable.xml
 - servo1.xml
 - servo2.xml

— Snap!Mobile

- arduino
 - log
 - s2a_fm_debugging.log
 - PyMata
 - __init__.py
 - pymata_command_handler.py
 - pymata.py
 - pymata_serial.py
 - serial
 - tools
 - __init__.py
 - list_ports_linux.py
 - list_ports_osx.py
 - list_ports_posix.py
 - list_ports.py
 - list_ports_windows.py
 - miniterm.py
 - urlhandler
 - __init__.py
 - protocol_hwgrep.py
 - protocol_loop.py
 - protocol_rfc2217.py
 - protocol_socket.py
 - __init__.py
 - rfc2217.py
 - serialcli.py
 - serialjava.py
 - serialposix.py
 - serialutil.py
 - serialwin32.py

```

├── sermsdos.py
├── win32.py
├── s2a_fm.py
├── scratch_command_handlers.py
├── scratch_http_server.py
├── xlate.cfg
├── StandardFirmataYun
├──   └── StandardFirmataYun
├──   └── StandardFirmataYun.ino
├── wiredMotorTest.xml
├── yun_blink.xml
├── yun_motor_control.xml
├── yun_motor_control.xml~
├── blink.xml
├── s2a_fm_Snap_base_dutch.xml
├── s2a_fm_Snap_base_Es.xml
├── s2a_fm_Snap_base_FR.xml
├── s2a_fm_Snap_base.xml
├── s2a_fm_Snap_base_zh_cn.xml
├── s2a_fm_Snap_base_zh_tw.xml
├── spinning_sprite.xml
├── Turn On LED On Pin 6.xml
├── license.txt
├── README.md
├── s2a_fm.py
├── scratch_command_handlers.py
├── scratch_command_handlers.pyc
├── scratch_http_server.py
├── scratch_http_server.pyc
├── xlate.cfg

```

11. 參考資料

Arduino

<http://arduino.cc/>

Snap!

<http://snap.berkeley.edu/>

Scratch

<http://scratch.mit.edu/>

Arduino Standard Firmata

<http://playground.arduino.cc/Interfacing/Firmata>

PyMata

<https://github.com/MrYsLab/PyMata>

NewPing Arduino Library

<https://code.google.com/p/arduino-new-ping/>

12. 問題、意見及錯誤回報

請聯絡 MisterYsLab@gmail.com。