



TENCODE TECHNOLOGY

# 2021 七賢水果辨識營隊 Week01

## 水果辨識營隊

- 本營隊採用Arduino UNO開發板，以RGB感測器，進行水果甜度辨識，並且探究水果外表色彩與甜度之相關性，達到可生活實用之非破壞式量測!

# Week01課程安排(每整點休息10分鐘)

8:00-9:00 水果甜度與外觀相關性

9:00-10:00 Arduino歷史基本認識

10:00-11:00 下載Arduino編譯器並撰寫簡易程式

11:00-12:00 讀取光敏電阻之值，並且使單色LED發光

12:00-13:00 午休

13:00-14:00 設計RGB三通道的實體遮罩並量測

14:00-15:00 將量測到之結果紀錄，並以RGB LED燈使其發光

15:00-16:00 統計與線性回歸

16:00-17:00 以Excel實際測試



TENCODE TECHNOLOGY

# 第一堂課程

## 水果甜度與外觀



TENCOD TECHNOLOGY

#天閣科技 十行程式碼改變你對程式的想像

# 第一堂課程-水果甜度





# 第一堂課程-水果甜度-表面有皺褶





# 第一堂課程-水果甜度-深淺變化

## 香蕉放七天營養變化 大家最喜歡吃哪種呢?

香蕉放置一段時間後，顏色會一直變化，口味不同之外，營養素也大不同。  
C:碳水化合物/S:總糖量/Fru:果糖/Fb:膳食纖維

第0天



第1天



第3天



第7天



竟然放久熱量更低!

熱量高

熱量低

熱量:90Kcal  
C:23.3g  
S:9.2g  
Fru:2.8g  
Fb:2g  
VitA:44IU  
VitC:11.5mg

熱量:87Kcal  
C:22.2g  
S:13g  
Fru:4.3g  
Fb:1.7g  
VitA:46IU  
VitC:6.4mg

熱量:84Kcal  
C:21.3g  
S:11.9g  
Fru:4.5g  
Fb:1.7g  
VitA:0IU  
VitC:4.8mg

熱量:68Kcal  
C:17.1g  
S:10.8g  
Fru:4.5g  
Fb:1.6g  
VitA:0IU  
VitC:4.3mg

每100g計算營養素

Instagram YouTube 營養師杯蓋

資料來源:食品成分資料庫



# 第一堂課程-水果甜度-光澤變化

## 挑選櫻桃的4個重點



### POINT 1 果梗硬

從櫻桃梗看是否新鮮  
顏色鮮綠且形狀完整的梗蒂，表  
示新鮮且沒有遭到碰撞



### POINT 2 硬度、彈性

彈性佳、果肉結實的櫻桃  
是新鮮的象徵，水分也比較充足。



### POINT 3 光澤潤

櫻桃的採收、包裝及運輸都必須在短  
時間內完成，維持表皮的光澤；  
若是儲運過程不佳、陳列過久，都會  
破壞光澤、顯得黯淡，甚至讓果肉脫  
水，導致表皮皺摺。



### POINT 4 根蒂凹、顏色深

根蒂處形狀愈凹、顏色愈深  
表示甜度較高。



## 第一堂課程-水果甜度-顏色變化



較成熟的蜜棗顏色會越黃，甜度較高但是口感鬆軟；反之，顏色越綠的蜜棗有較脆的口感，但是甜度就較低。前者因成熟度高，所以不耐久放，應盡早食用。

挑選時要注意果蒂周圍是否凹陷寬廣，光滑平順無高低不平的皺摺，若果蒂周圍突起者就代表品質較差。



- 課堂練習時間:15分鐘!
- 請各位上網搜尋，水果甜度的關鍵判斷要素是什麼？
- 三人一組，各組不同水果，進行上台分享!

# 第一堂課程-水果甜度-科展

中華民國第 59 屆中小學科學展覽會  
作品說明書

---

國小組 化學科

080203

揭開柳橙甜度的秘密

學校名稱：高雄市三民區東光國民小學



# 第一堂課程-水果甜度-科展

## (一) 甜度：

人對甜味的喜愛是出於本能；在沒有人工的化學甜味劑出現之前，糖是最主要的甜味來源。甜度可以說是糖溶液所呈現出的甜味感覺程度。目前真正的甜度尚無法用儀器分析，而是靠官能品評來打分數。

糖濃度愈高，甜度愈大，這是不爭的事實；但是在糖溶液中若有其他溶質的存在，也會影響甜味的感覺，例如糖中有酸、水果抹鹽、果蔬汁加少量的鹽，均有助於甜度的加強。因此甜度的大小必須在僅由糖提供甜味的情況下，才能說它與糖度的多寡有關。又糖的種類不同時，甜度也不一樣，例如葡萄糖不如蔗糖及果糖甜。溫度不同時，甜度的表現也會不同，例如葡萄糖與果糖在冷水中的甜度比熱水中甜度大。水果及果汁飲料中的糖，往往不是單純的一種糖，因此甜度與糖度之間還有一些差距。



# 第一堂課程-水果甜度-科展

## (二) 糖度：

目前國人通稱的糖度，和蔗糖工業上所用的糖度略有不同。在果實、果汁、食品品質分析上所稱的糖度通常是以 **Brix 度數** 表示，指的是用比重計法測定樣本溶液的比重。**Brix 度數** 雖然習慣上看成是含糖重量百分率(一般果汁中存在的糖類如葡萄糖、果糖與蔗糖的比重或折射率相似。)然而實際上所代表的意義，應該是指果汁所含溶解物概略重量百分率。一般的水果及水果產品，若不含不溶性物質(如果汁)時，其可溶固形物含量即以糖度折射計法所得到的讀數表示

# 第一堂課程-水果甜度-大學研討會

SEAIT2017 第 9 屆企業架構與資訊科技研討會

## 水果成熟度分級檢測系統之設計

張正弘 老師

李權容 學生

藍家翔 學生

德明科大

德明科大

德明科大

資訊科技科

資訊科技科

資訊科技科

[lion@takming.edu.tw](mailto:lion@takming.edu.tw)

[penda@gmail.com](mailto:penda@gmail.com)

[penda@gmail.com](mailto:penda@gmail.com)



# 第一堂課程-水果甜度-大學研討會

## 摘要

目前台灣的水果自動化分級技術雖然已發展完全，但成本仍居高不下，一台桌上型的機器就要價95萬元，更不要提價格高達3000萬元的線上型分級器，所以若是我們能夠做出價格低廉，機動性高，檢測成果穩定的水果自動化分級器，相信能在農產的市場裡打出一片天。水果自動化分級器在農產的市場裡絕對是不可或缺的，一直以來消費者在購買水果時的共同疑問就是水果夠不夠甜，而這種主觀又抽象的問題也總是得不到準確的答案，要是我們能夠讓水果廠商都購買一台水果自動化分級器的話，有了統一且標準的甜度指標，對消費者來說相信也會更有說服力。

本研究的貢獻於果農，方便水果商或農民可以不用再以拍打或者是其他方式來辨別水果成熟程度、水果種類或者是甜度等等，便於分類、分級。我們主要以影像處理方式及電壓晶體辨別水果色澤及瑕疵、檢測水果甜度、判斷大小、重量辨別。在這個科技時代以機器對水果實施非破壞性個體檢驗，除可提高品質、增加產品經濟價值、確保果農收益外，將有助於提升消費者購買信賴，藉由機器分析數據，診斷果農栽培管理方法的妥適性，除了可降低人成本外更能提高水果品質。



# 第一堂課程-水果甜度-大學研討會

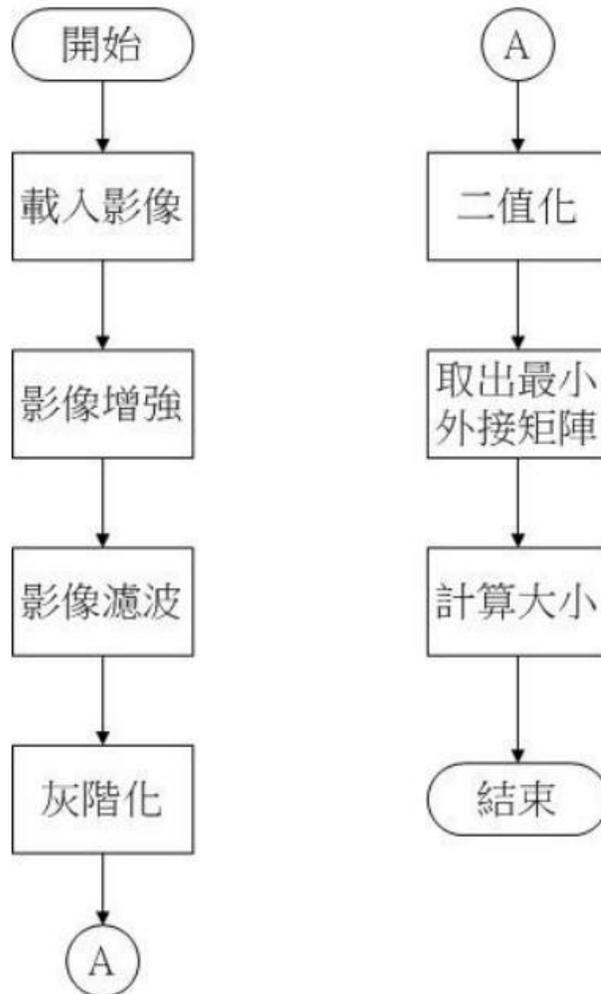
## 4. 實驗設計、過程及結果

初期實驗的目標是找出水果的相對大小，並藉以分類水果的級數，一般是越大的水果等級較高。我們先載入靜態的水果原始彩色影像，此處選擇檸檬為實驗的水果，處理的過程包含先進行影像的前處理，前處理的步驟為影像的增強及濾波，待影像品質改善後，先將影像灰階化，再將灰階轉二值化，二值化的閾值採用Otsu's的方法決定。未來也可以在二值化步驟後標示出水果的瑕疵處，接著就可以計算水果的大小。圖9中為處理的流程圖，並附上部分處理過程的程式碼。

水果的大小計算可以由三個方式來進行

- 直接由二值化影像來計算圖中的白色點的數目，超過某的數字或比例及可以定義為大的水果。
- 求取外接最小矩形框，並讀取矩形的長、寬資料，以長、寬中的最大值為水果大小的判斷數值，外接矩形的圖示如圖13。

# 第一堂課程-水果甜度-大學研討會





# 第一堂課程-水果甜度-大學研討會

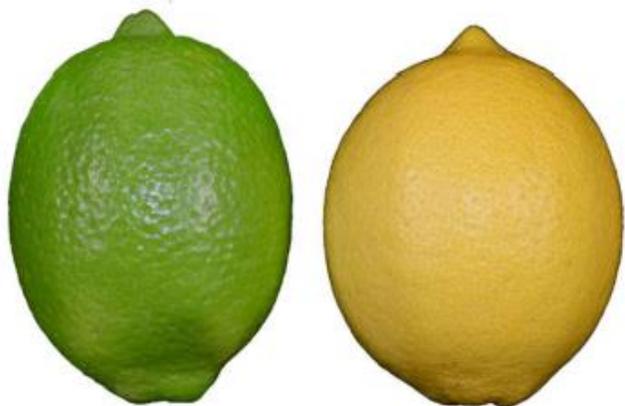


圖10 檸檬原圖

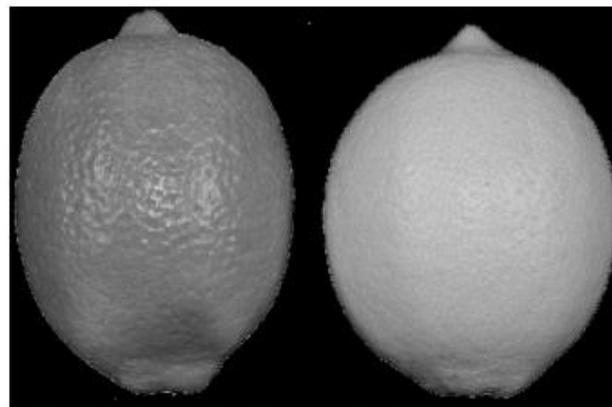


圖11 灰階化後的檸檬圖



圖12 二值化後的圖



圖13 外接最小矩形框





# 第一堂課程-水果甜度-大學研討會

## 5. 討論及未來工作

本論文依據水果的影像特性進行水果大小的判定及瑕疵的檢測，依前章的實驗結果，目前所做出的結果已經可以達到初步判讀大小的數據。但是水果影像存在著非常多的變異性，例如拍攝的角度及水果本身的光澤造成反光等，處理過後的影像必定存在非常多的雜訊，此雜訊會影響二值化的結果，並改變外接矩形的大小。未來將繼續研究影像增強及影像濾波等技術，增強影像對比，消除雜訊，並考慮以形態學濾波進行二值化的進接觸，以達到最佳影像。未來也希望可以繼續做出瑕疵檢測效果及水果品種分類。

• 下課休息10分鐘



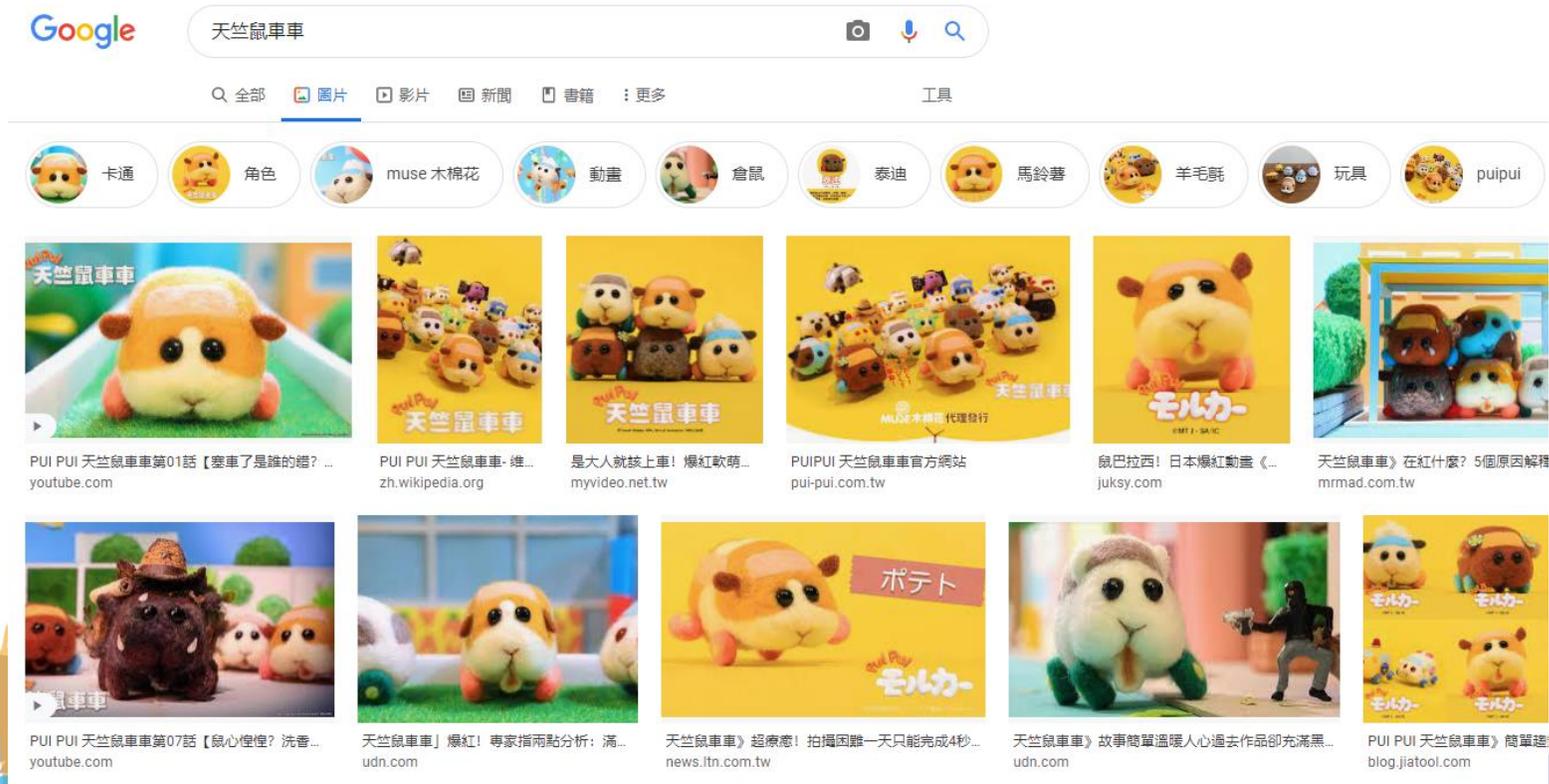
TENCODE TECHNOLOGY

# 第二堂課程

## Arduino介紹

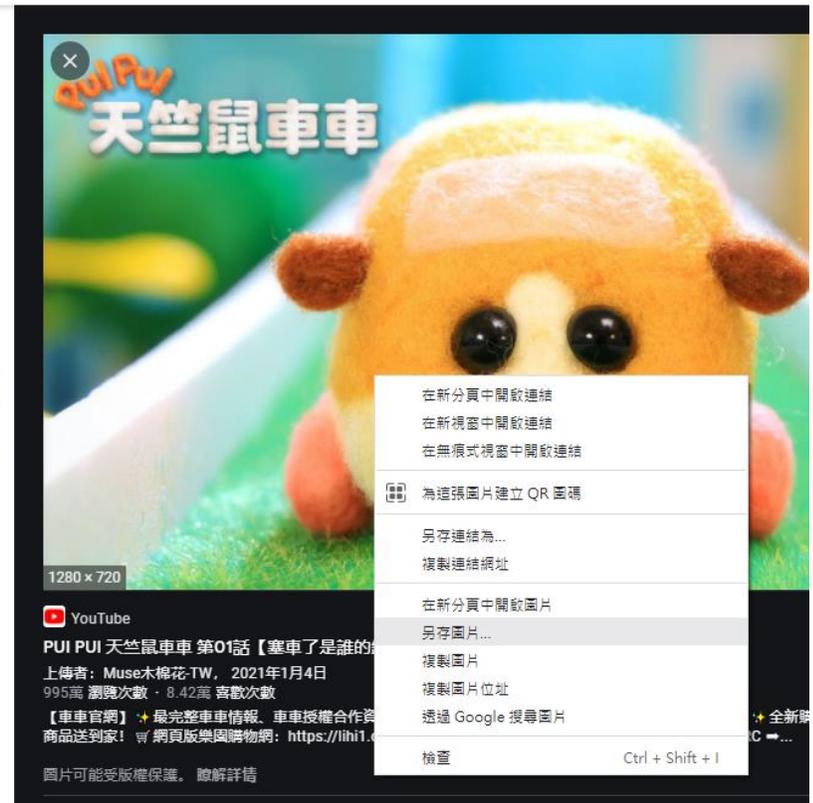
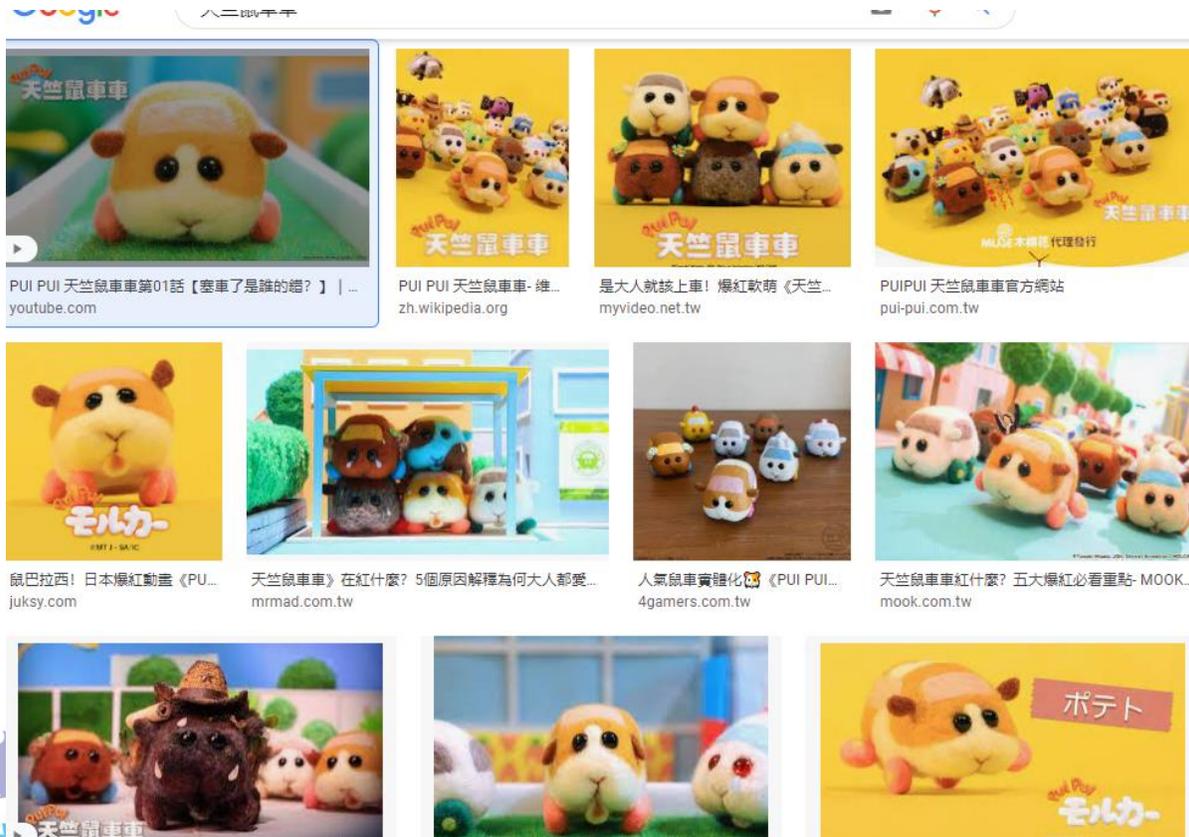
# 第二堂課程-先認識圖片

- 請到google找一張你喜歡的圖片。



# 第二堂課程-先認識圖片

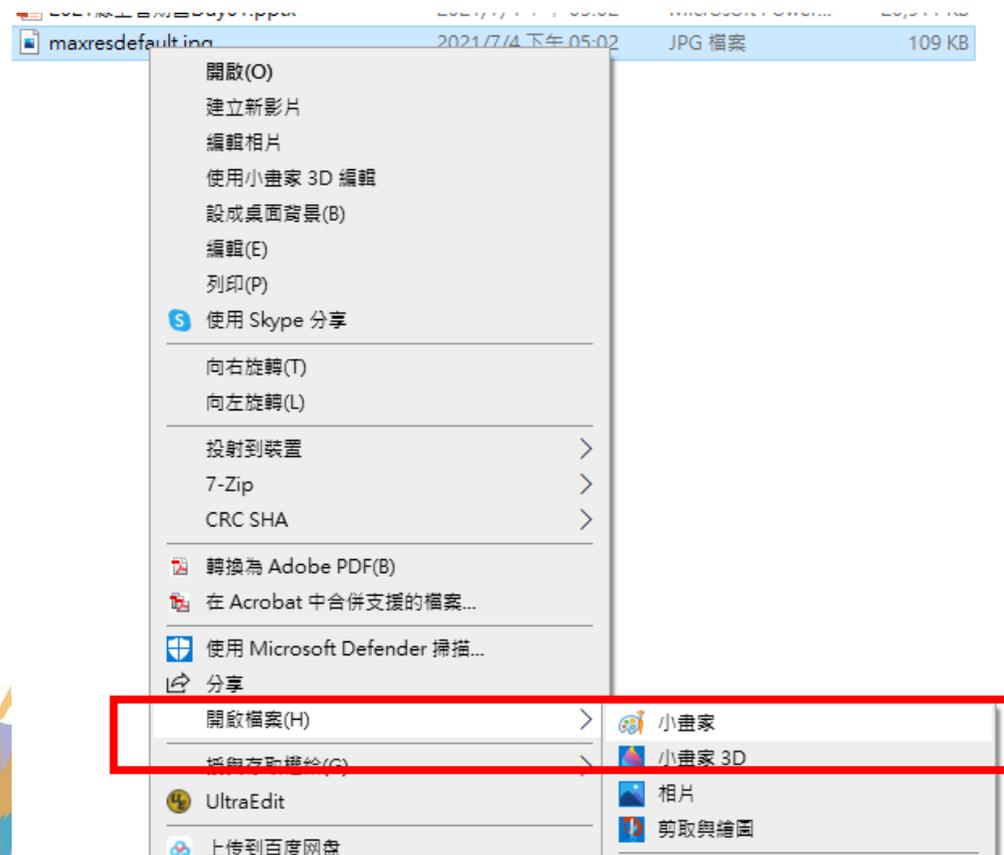
- 將圖片下載下來!!





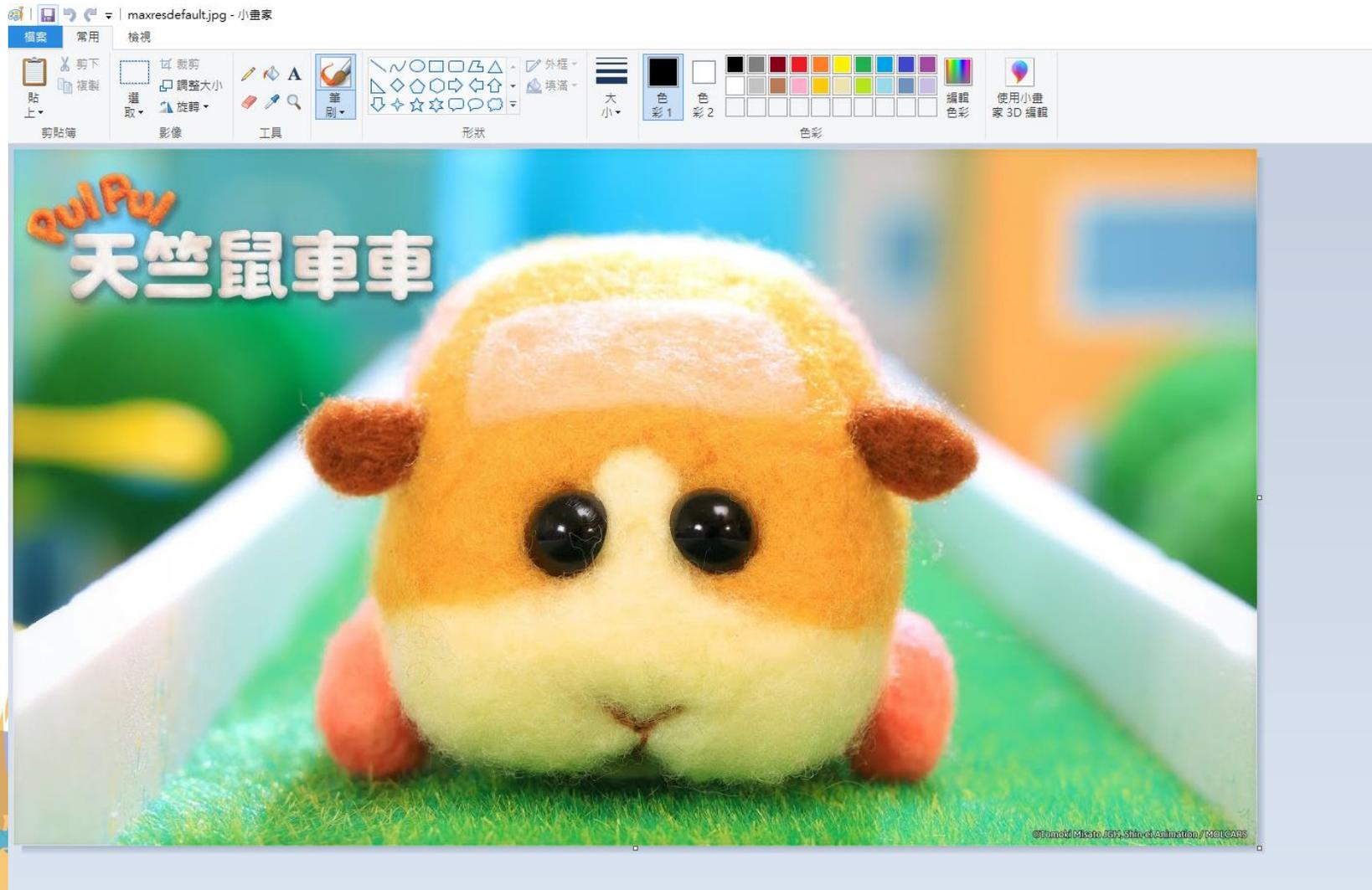
# 第二堂課程-先按右鍵->開啟檔案->選小畫家

- 利用小畫家把圖片打開!!





# 第二堂課程-小畫家打開圖片





## 第二堂課程-將圖片放大~

這些一格一格就是圖像的最小單位。  
->稱為像素，英文是Pixel。



● 課堂練習時間:5分鐘!

## 第二堂課程-Pablo Ruiz Picasso



**Pixel**  
**=Picture + Element**

# Arduino

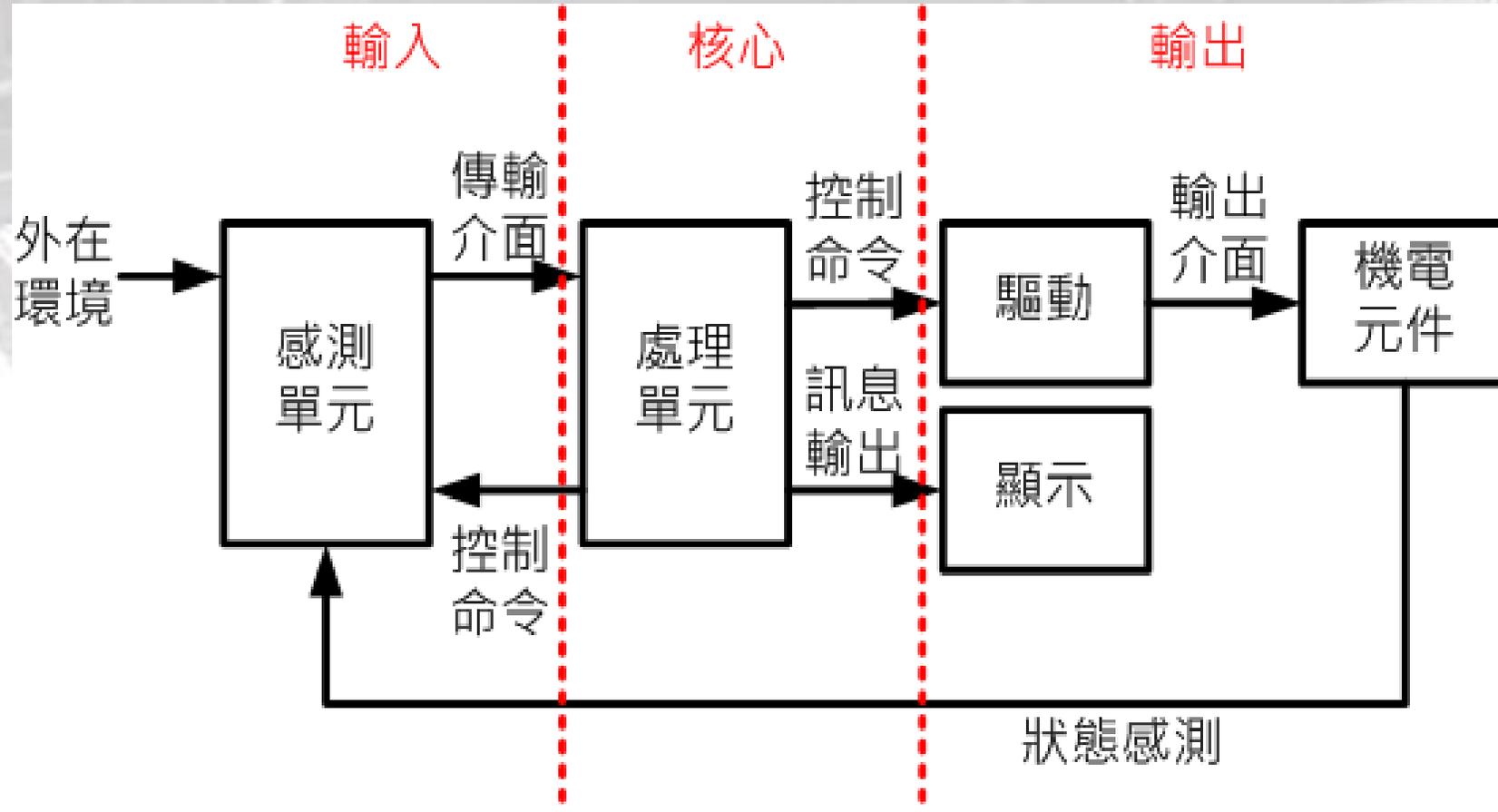
## 發展歷史及簡介

# 物理運算平台

- Physical Computing
- 廣義來說包含了感測外在環境的訊息、與使用者溝通的人機介面、能夠對應命令產生相對動作的致動器，最後最重要的是需要一個處理核心，在這邊通常是指單晶片微控制器。
- 藉由這些系統元素的組合搭配，可以完成一個只需要開起電源後便可**獨立運作執行**的系統。

# 物理運算平台架構

整個系統平台可以分3個部分：輸入、核心和輸出

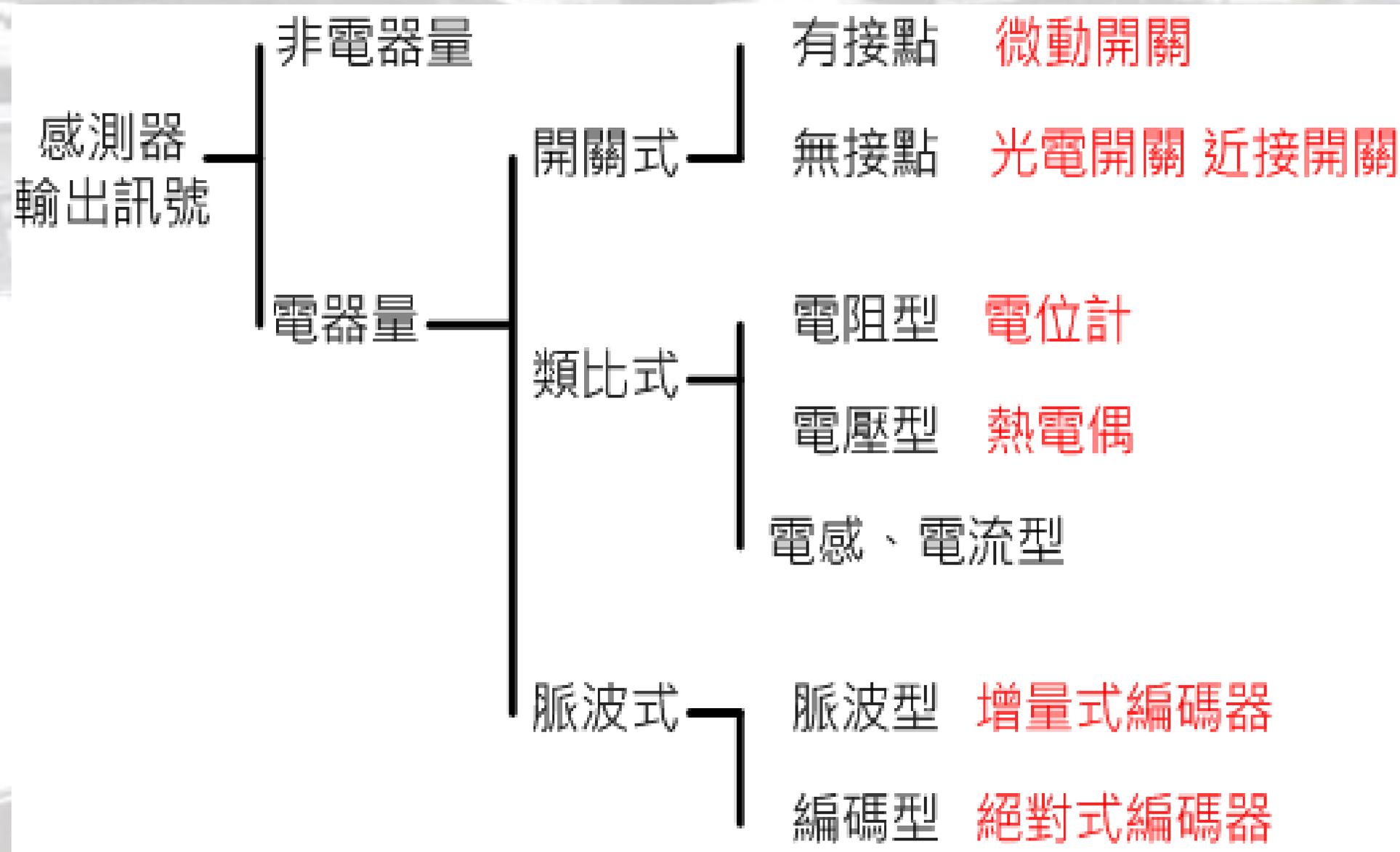


# 輸入

- 依照需求使用某些裝置對系統下達指令。
- 針對特殊功能的需要，對於物理量進行量測，例如溫度、壓力、形變等，再將這些物理量轉換為電壓、電流、電感等變化；甚至對於輸出端的機械系統進行監控等。

感測器類型	待測量	轉換訊號
浮球	液面高度變化	位移
熱敏電阻	溫度	電阻變化
光敏電阻	光線	電阻變化
加速度計	加速度	電壓
霍爾IC	磁場	電壓
壓力計	壓力	電阻變化

# 簡易感測器分類



# 核心

- 就像我們人類的腦袋，它可以感受到耳朵聽到的聲音、感覺肌膚受到的碰觸，它也可以控制手臂的擺動、腳步的運動，他是人類一切活動的根源，不能沒有腦的思考。
- 物理運算平台的指揮官就是單晶片，如果系統比較複雜，也有使用電腦為核心。

# 輸出

輸出可以是一個小螢幕、一顆燈泡或是一台機構複雜的設備，當設計的人希望這個系統要以怎麼樣的方式呈現，輸出的種類就隨之更換，有的也許講求功能性，有的卻需要能吸引目光。

# 話說從頭-從核心談起

- 核心單元以單晶片為主，單晶片(或稱微處理器，大陸則為單片機)為核心的嵌入式系統。
- 某些場合會將單晶片系統視為比較低階的應用，採用的晶片主要多為**8位元**或**16位元**，而嵌入式系統視為高階場合應用，核心多為系統晶片或是**32位元**的處理器為主。
- 單晶片系統的運算速度大約為**20MHz**左右，經由倍頻的技術可以提升晶片的處理速度。

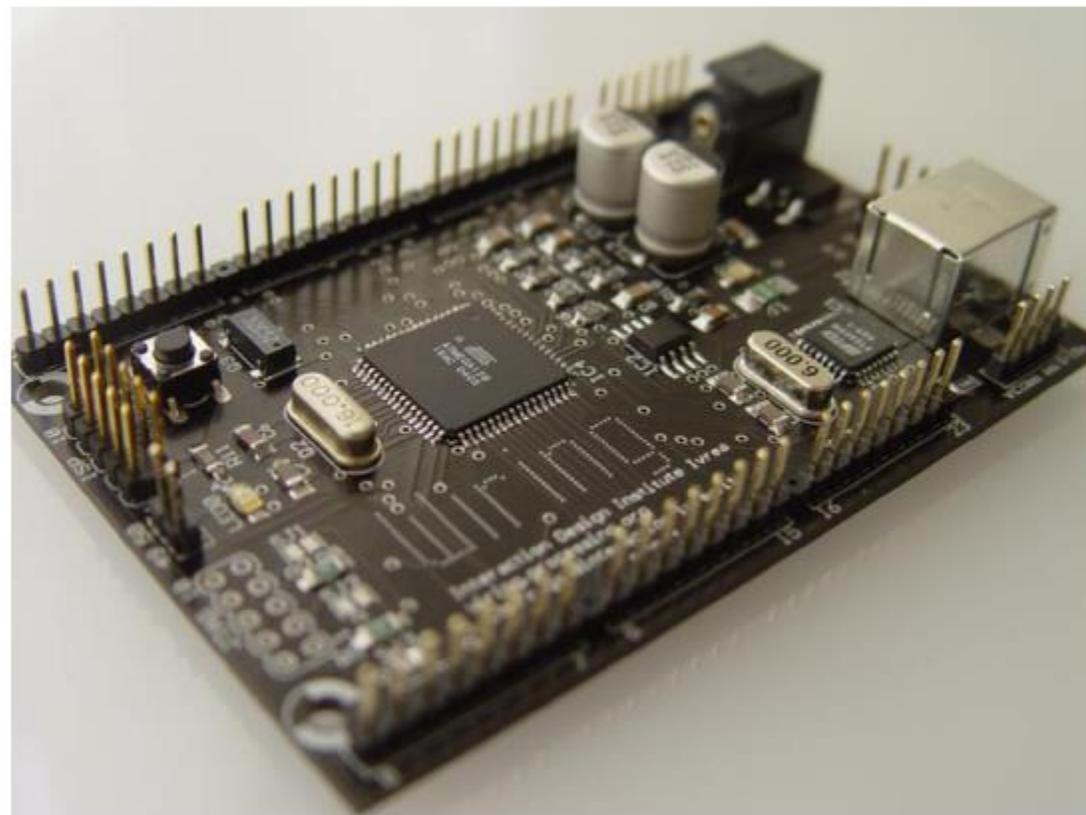
# 話說從頭—從核心談起

- 單晶片為主的系統，單位時間內只能處理一件事情，更進一步解釋是只能執行單行程式碼。
- 所有任務必須依照已排定的優先順序來執行，故功能上的安排會較為單純，一般用於較為簡潔的系統，或是需要有一定固定順序執行的設備。
- 內部程式都是預先設計撰寫完成，除非做系統的更新改版，這種程式我們稱為**韌體**，主要負責軟體與硬體之間的溝通。

# Arduino發展歷史及簡介

**2005年**時，任教於**義大利**北部伊夫雷亞(Ivrea)一所互動設計學院(**IDII**, Interaction Design Institute Ivrea)的Massimo Banzi和David Cuartielles教授，希望能替學生及互動設計師，找到一種能幫助他們學習電子和感測器基本知識，並快速地設計、整合互動作品的原型(Prototype)的微電腦裝置。然而這是一項物理計算(Physical Computation)或者稱為物理互動設計(Physical Interaction Design)，有鑑於當時市面上的微電腦控制相關產品所費不貲，有些產品採用的**程式語言深奧難懂**，也**不適合設計學院**的學生使用，所以找來幾位志同道合的夥伴及學生(如下頁圖)，以**11世紀北義大利的一個國王Arduino**為名(同時此字具有強而有力的意思)，設計出開放式(Open Source)微電腦控制板以及開發工具。

# 發展歷史及簡介



Wiring I/O Board

Hernando Barragán | Interaction Design Institute Ivrea | June 2004

h.barragan@interaction-ivrea.it

## Wiring: Prototyping Physical Interaction Design

### Thesis Committee

**Massimo Banzi** . Associate Professor

*Primary advisor*

**Casey Reas** . Visiting Assistant Professor

UCLA Design | Media Arts,

*Secondary advisor*

**Gillian Crampton Smith** . Director

**Andrew Davidson** . Chair of the Academic Programme

圖1.1 2003/The Untold History of Arduino by **Hernando Barragán** (資料來源：<http://arduinohistory.github.io/>)

# 發展歷史及簡介



**David Mellis**

**Tom Igoe**

**Massimo Banzi**

**Gianluca Martino**

**David Cuartielles**

# 發展歷史及簡介

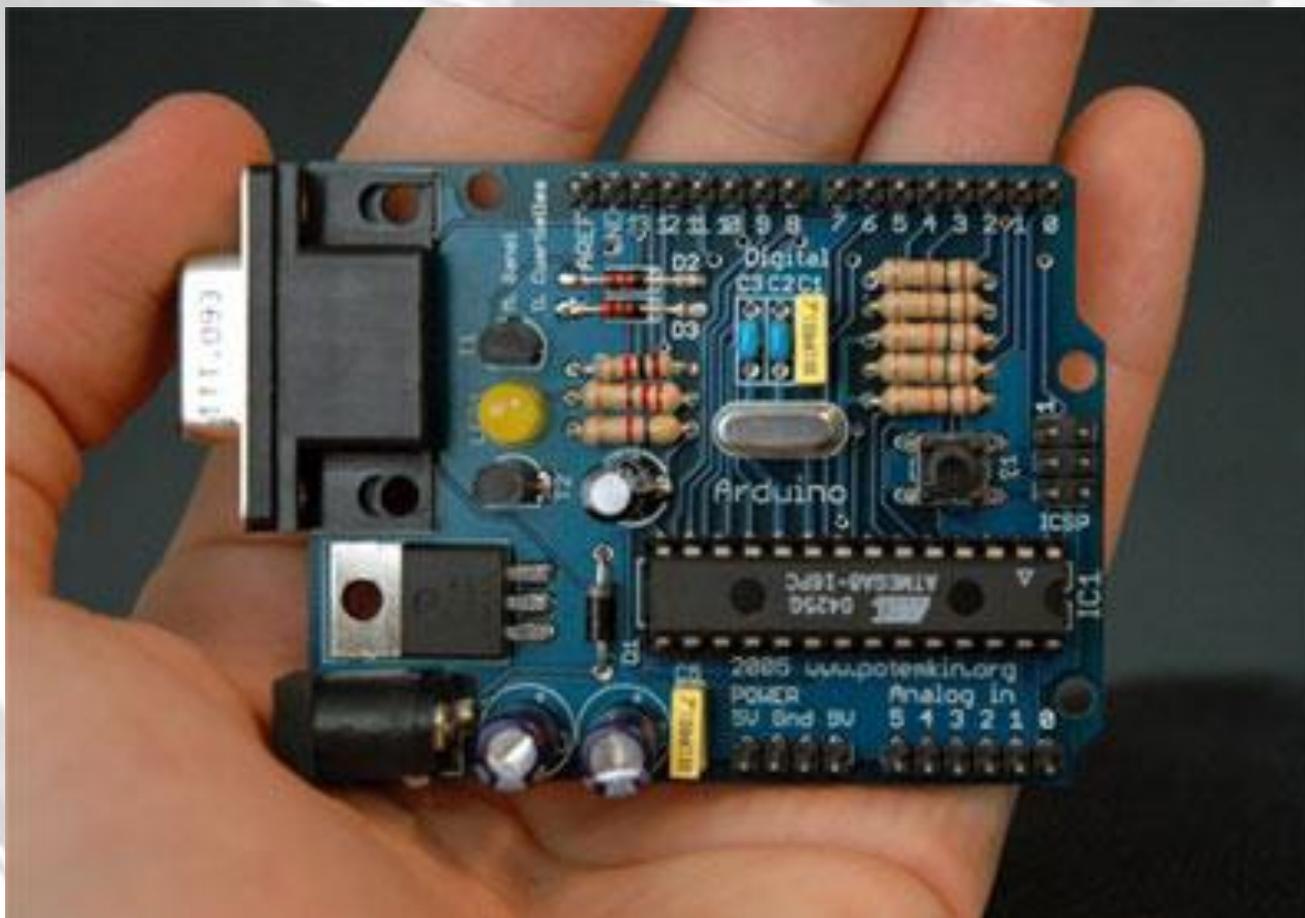
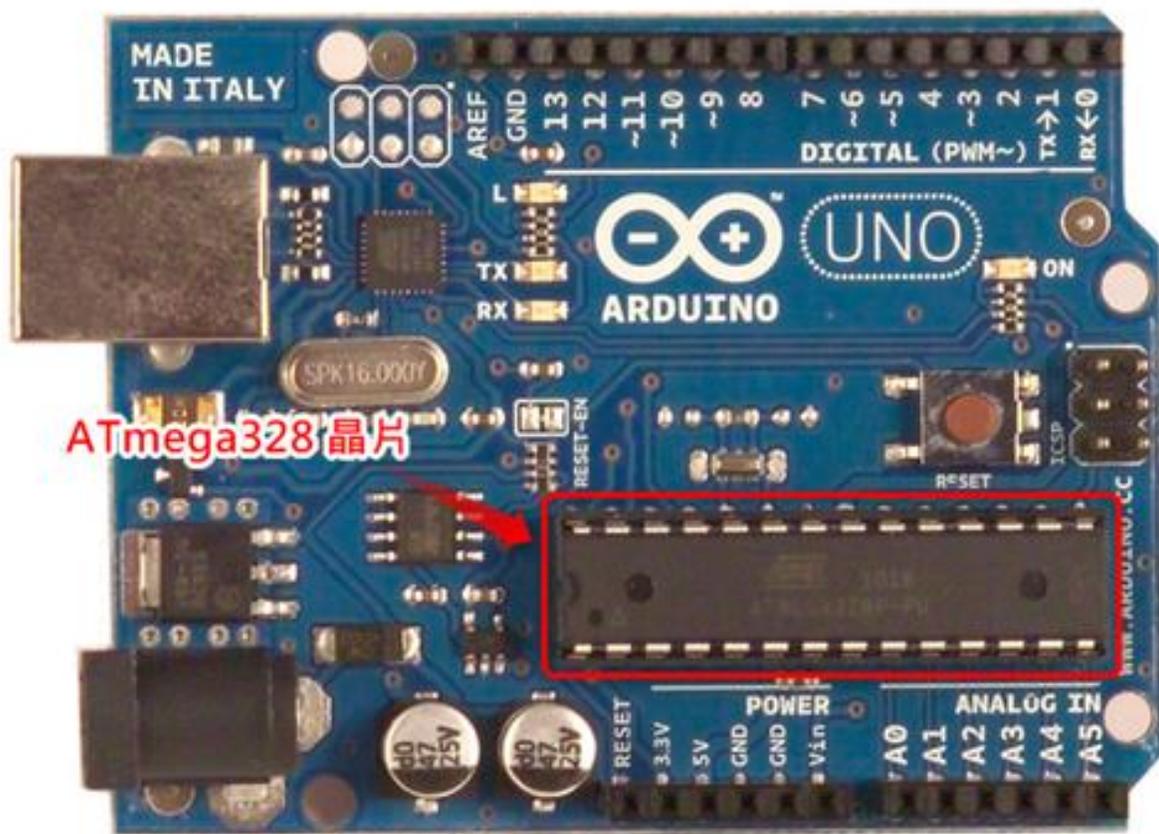


圖1.1 2005/3設計的Arduino控制板（圖片來源：<http://www.arduino.cc/en/Main/ArduinoBoardSerial>）

# 發展歷史及簡介



# 發展歷史及簡介



## PDIP

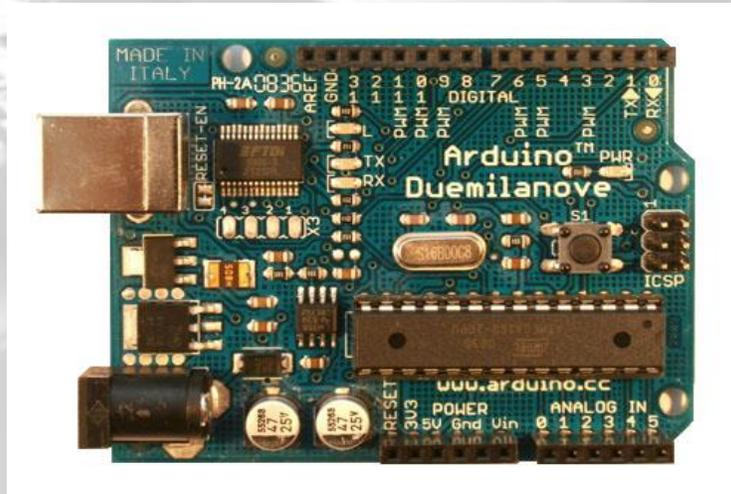
(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 ( $\overline{SS}$ /OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

# 發展歷史及簡介

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega48A	4KBytes	256Bytes	512Bytes	1 instruction word/vector
ATmega48PA	4KBytes	256Bytes	512Bytes	1 instruction word/vector
ATmega88A	8KBytes	512Bytes	1KBytes	1 instruction word/vector
ATmega88PA	8KBytes	512Bytes	1KBytes	1 instruction word/vector
ATmega168A	16KBytes	512Bytes	1KBytes	2 instruction words/vector
ATmega168PA	16KBytes	512Bytes	1KBytes	2 instruction words/vector
ATmega328	32KBytes	1KBytes	2KBytes	2 instruction words/vector
ATmega328P	32KBytes	1KBytes	2KBytes	2 instruction words/vector

# Duemilanove

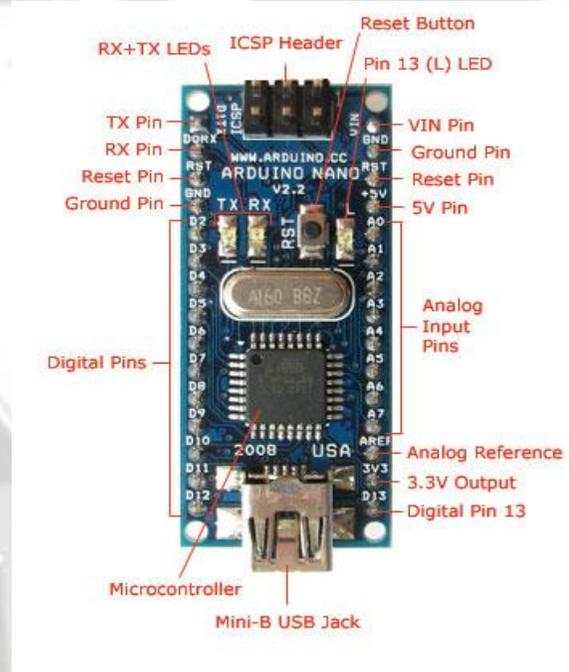
- Arduino標準版本，目前最新的標準版名稱定為UNO。
- 有許多針對此版本所發展出來的應用擴充版。預載核心為ATMEGA168或ATMEGA328。



圖片來源及所有權：Arduino.cc

# Nano

- 主要用於與麵包版結合使用。
- 晶片封裝與Duemilanove不同。



圖片來源及所有權: Arduino.cc

# MEGA

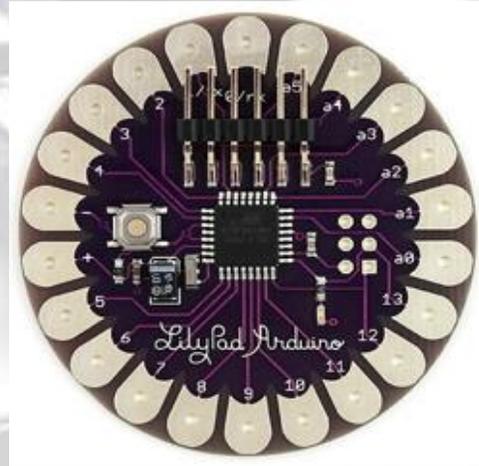
- Arduino的巨大版，包含了54個數位I/O，4組個UART，16個類比腳位，14組的PWM脈波訊號。



圖片來源及所有權：Arduino.cc

# LilyPad

- 以可以搭配在衣服上為設計主軸，故以圓形(鈕扣)為設計外觀，希望讓衣服與人可以有更多的互動關係。



圖片來源及所有權: [Arduino.cc](https://www.arduino.cc)

# Seeeduino

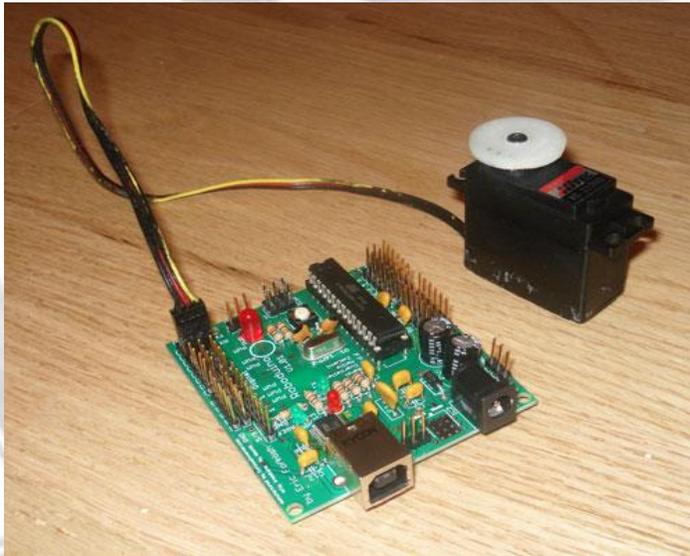
- 主要核心採用的不同的封裝，與電腦通訊的部分也使用了mini USB減少體積，開發完整相容於原廠的版子。



圖片來源及著作權：Seeeduino.com

# Roboduino

- 硬體上多了許多組伺服機要用的接線腳位，在使用多組以上的伺服機時，要考慮好電流大小的問題，避免版子無法動作或是損毀。



圖片來源及著作權：curiousinventor.com

# Funnel IO

- 結合了Xbee無線通訊模組，包含了對外的通訊介面，才可以適當表達目前的狀態。



圖片來源及著作權：funnel.cc

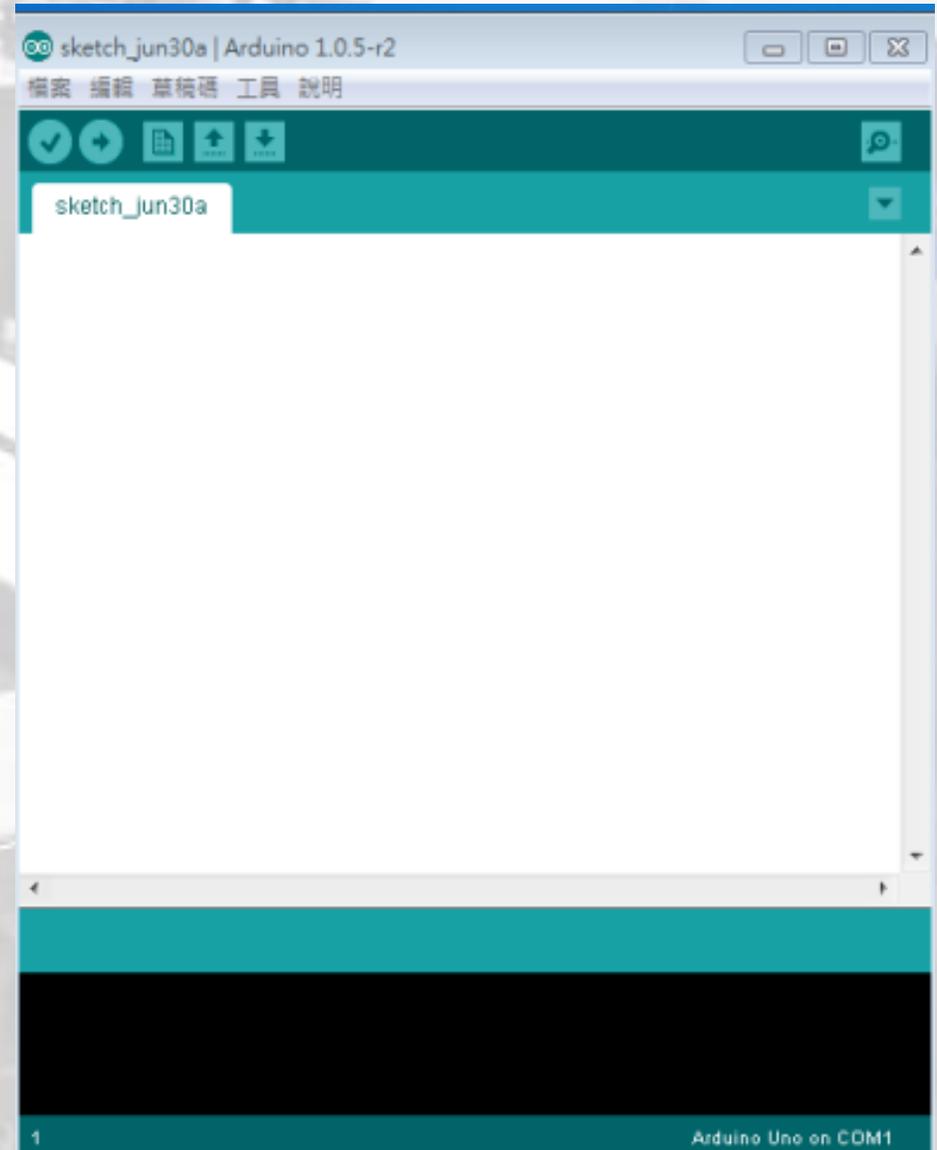
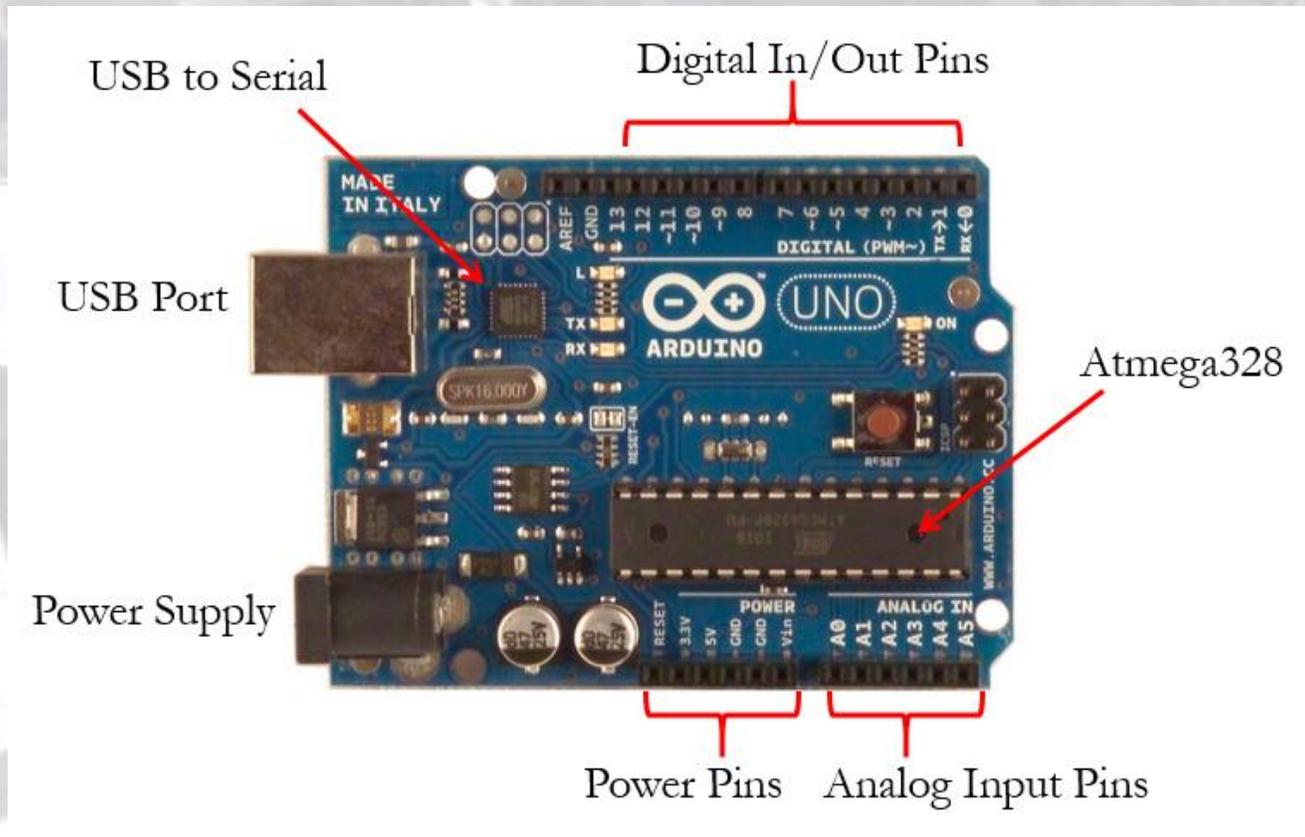
# 發展歷史及簡介

Arduino 是源自義大利的一個**開放源程式碼的硬體專案平台**，該平台包括一塊具備簡單I/O 功能的電路板(下頁圖)以及一套程式開發環境軟體(下頁圖)。

Arduino可以用來**開發交互產品**，比如它可以讀取大量的開關和感測器信號，並且可以控制電燈、電機和其他各式各樣的物理設備；Arduino也可以開發出與PC相連的周邊裝置，能在運行時與PC上的軟體進行通信。

Arduino的硬體電路板**可以自行焊接組裝**，也可以購買已經組裝好的模組，而程式開發環境的軟體則可以從網上免費下載與使用。

# 發展歷史及簡介



# 應用說明及思考

# 應用說明及思考

*Strangely Familiar.*

*Unusual Objects for Everyday Life.*

Applied Dreams Workshop | 10/22 January 2005

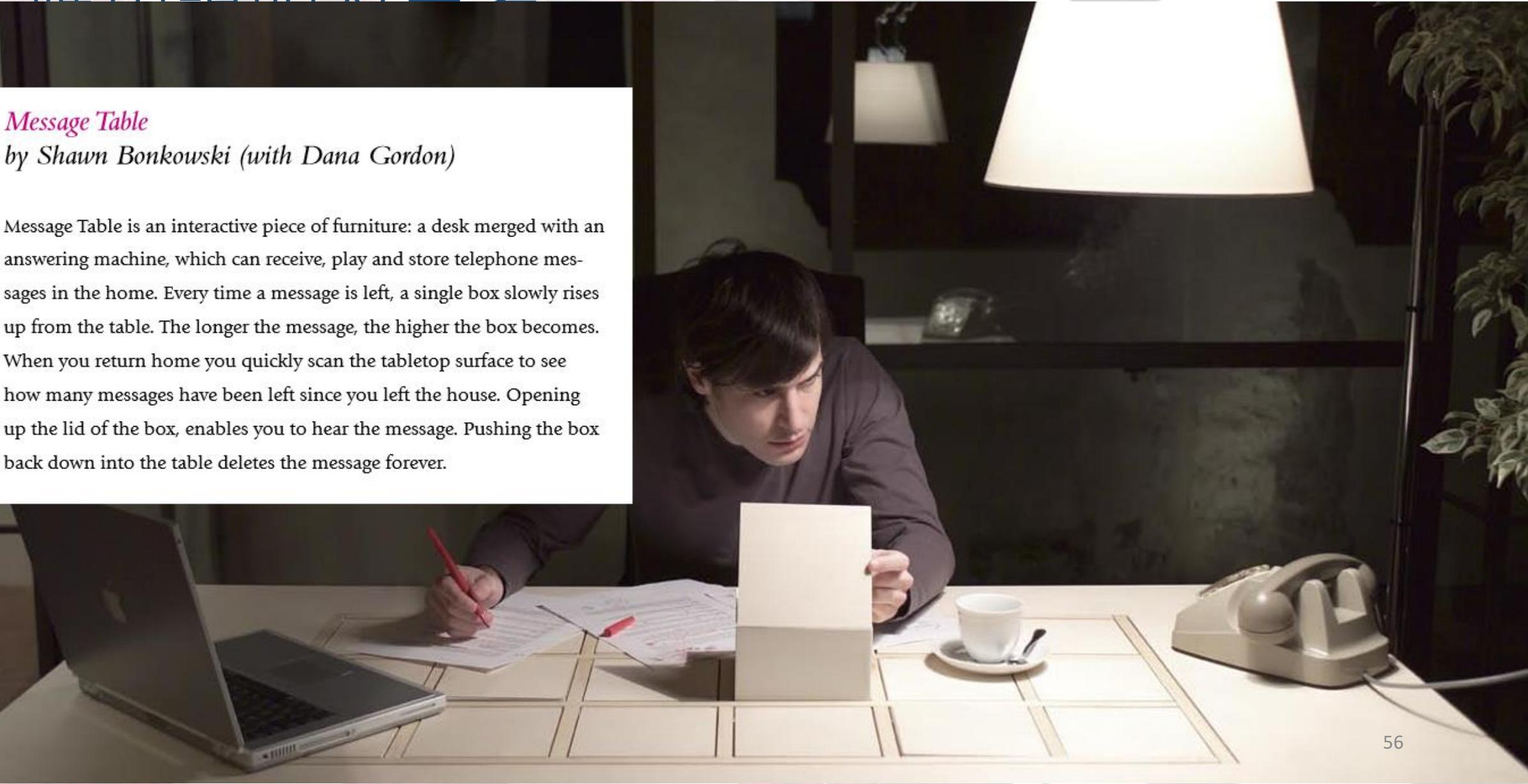
© Interaction Design Institute Ivrea

# 應用說明及思考

## *Message Table*

*by Shawn Bonkowski (with Dana Gordon)*

Message Table is an interactive piece of furniture: a desk merged with an answering machine, which can receive, play and store telephone messages in the home. Every time a message is left, a single box slowly rises up from the table. The longer the message, the higher the box becomes. When you return home you quickly scan the tabletop surface to see how many messages have been left since you left the house. Opening up the lid of the box, enables you to hear the message. Pushing the box back down into the table deletes the message forever.



# 應用說明及思考

## *Box of Sound*

*by James Tichenor (with David A Mellis)*

Box of Sound is a single radio station.

The exterior surface of the radio is made with thousands of rubber bands. When the user wedges an opening in the box the volume increases, but to the user the box appears to be in fact empty.

To play the radio louder, larger and wider objects have to be inserted.

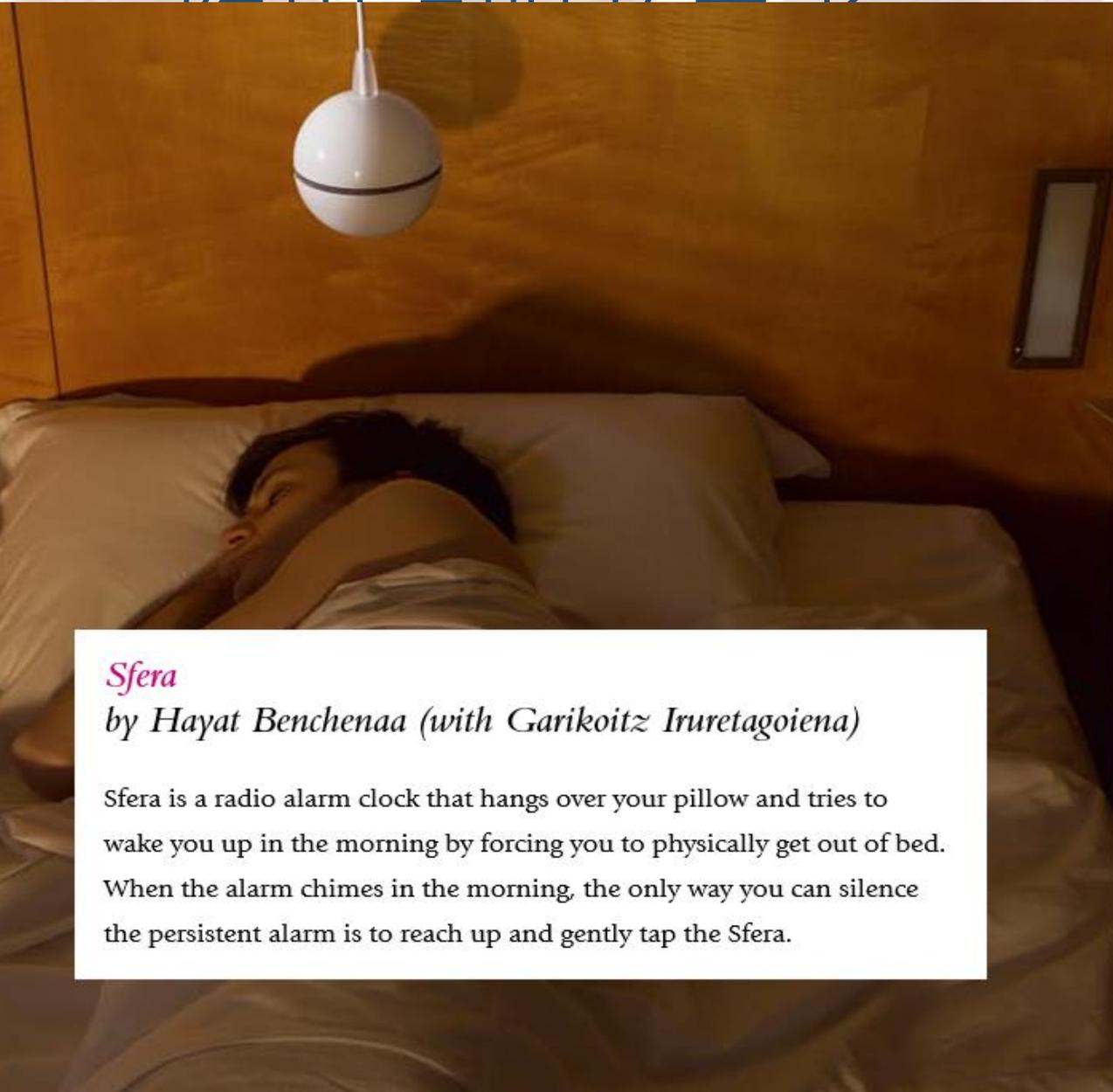


# 應用說明及田老

## *Feel the Music I*

*by James Tichenor and David A Mellis*

Feel the Music I is a radio with only a tuning knob. As the user spins the knob the sound of the station disappears, so only the knob itself indicates potential stations through tactile feedback. Phantom 'peaks and troughs' are physically felt through a force feedback mechanism placed in the radio. When the signal feels satisfying, the user lets go, causing the sound of the station to be heard.



*Sfera*

*by Hayat Benchenaa (with Garikoitz Iruretagoiena)*

Sfera is a radio alarm clock that hangs over your pillow and tries to wake you up in the morning by forcing you to physically get out of bed. When the alarm chimes in the morning, the only way you can silence the persistent alarm is to reach up and gently tap the Sfera.



# 應用說明及用者

## *Speak Out*

*by Tristam Sparks and Andreea Cherlaru  
(with Ana Camila Amorim)*

Phoning a friend who lives far away often feels the best way to catch up. But how many times have you phoned at a bad time? How many times have you felt a huge barrier of distance and silence as soon as you hang up? Speak Out is a way to keep an intimate link alive. Speak Out is an 'always-on' audio channel between you and your friend. It gives an ambient insight into what is happening in the other space. Effectively you are eavesdropping with permission.

# 藤田諒 田乃思孝



## *Tok Tok*

*by Aram Armstrong (with Haiyan Zhang)*

Tok Tok is a communication system that connects you to your distant loved ones in far away cities such as Tokyo, Toronto or Tel Aviv. Close partners each own a box. Knocking on one box transmits a sonar-

like pulse – heard as a duplicate knocking sound - to the ‘twinned’ box in the remote location. When the box receives this pulse it responds by transmitting the same pulse - or knocking sound – back to the original box. The time it takes for the sound to travel from Toronto to Tokyo and back again is based on the distance between the two cities. The further away the city is, the longer the delay.

# 應用說明及田老



## *Tug Tug*

by Haiyan Zhang (with Aram Armstrong)

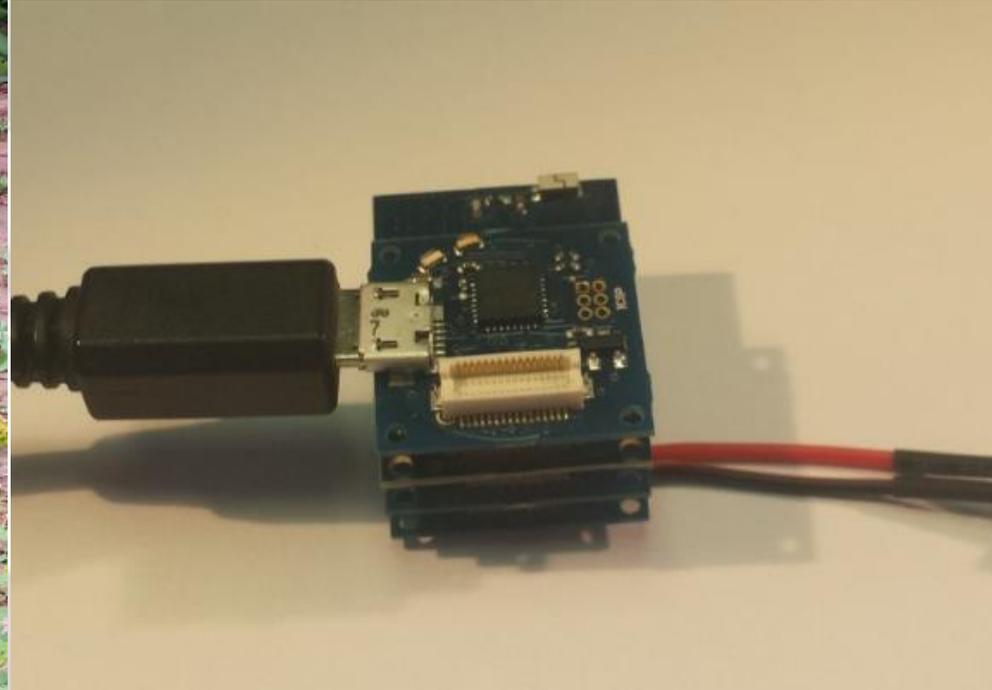
Tug Tug are dedicated telephones that offer an extra layer of interaction: the cord connecting the handset to the base on both phones forms a shared interactive object, allowing each person to physically affect the

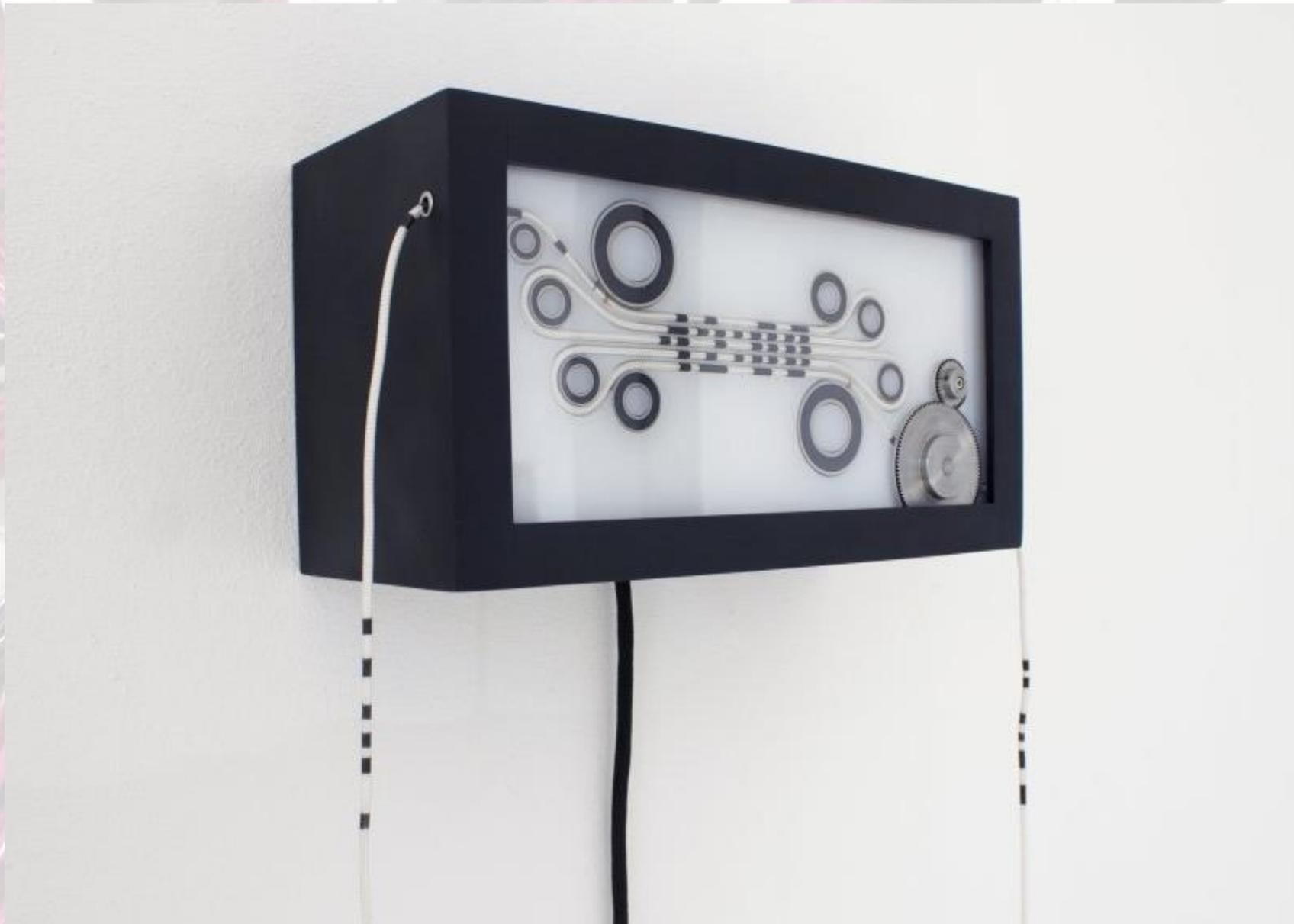


other object by pulling the cord.

This object starts to question what haptic nuances of communication could we gain from such physical interaction? How would you feel when your permissions are by-passed and someone just pulls your telephone straight off the hook – thereby creating a live audio link between both places.

# 應用說明及思考-GPS貓咪追蹤

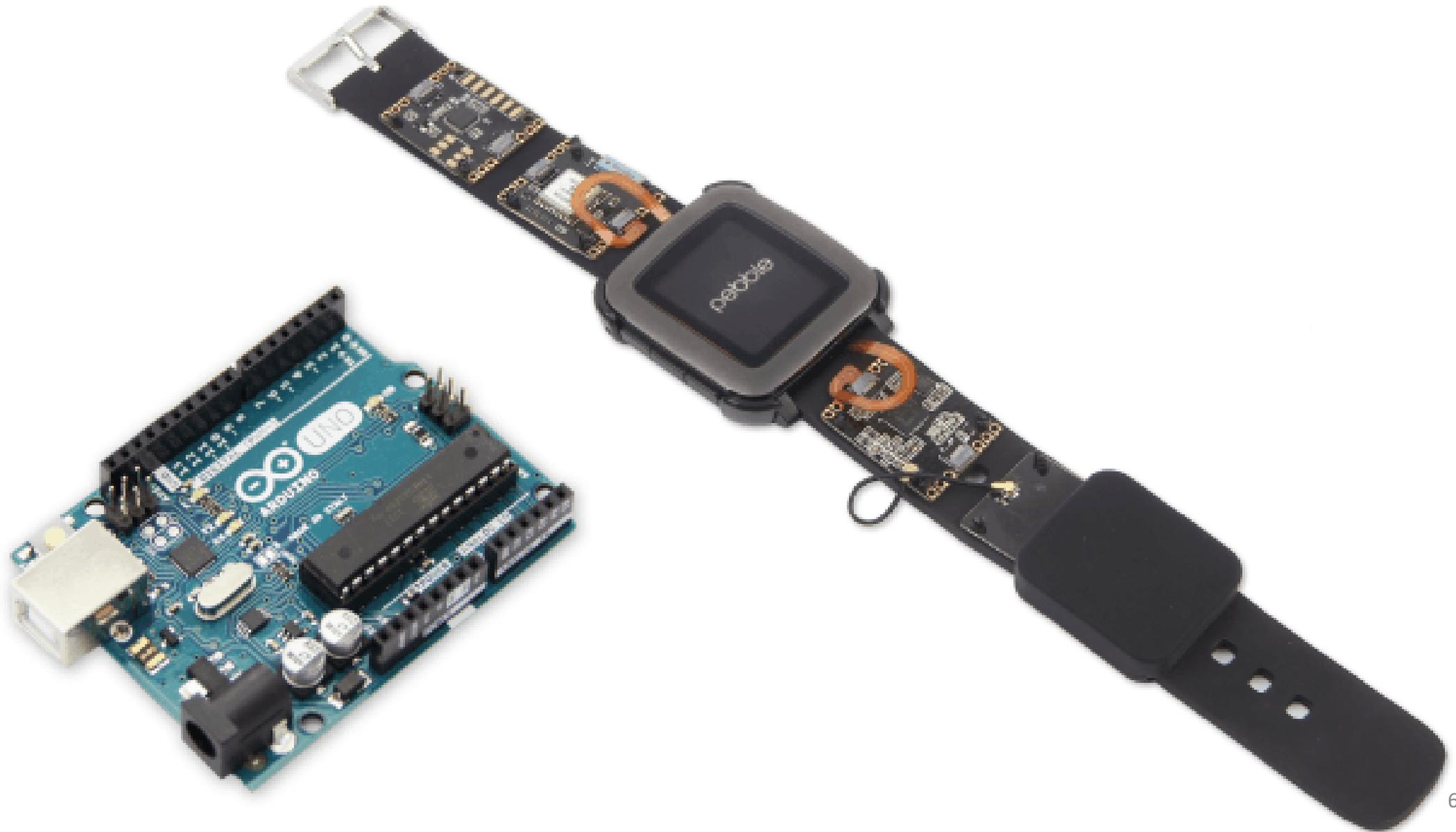




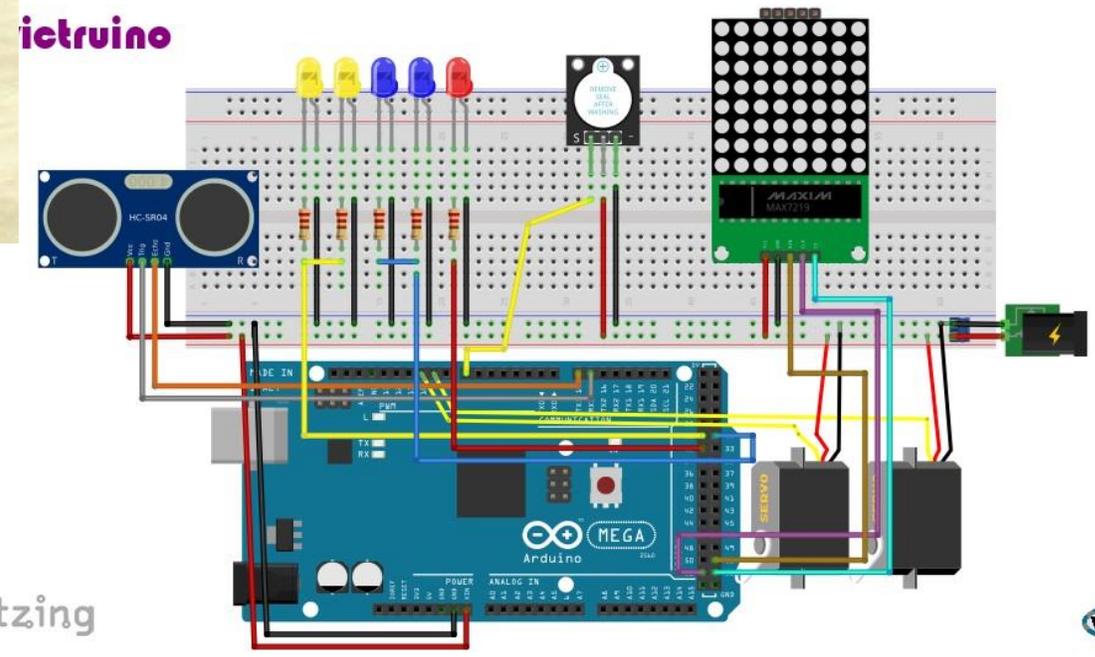
# 應用說明及思考-沒洗手警報器



# 應用於可穿戴裝置的 Arduino 開發板





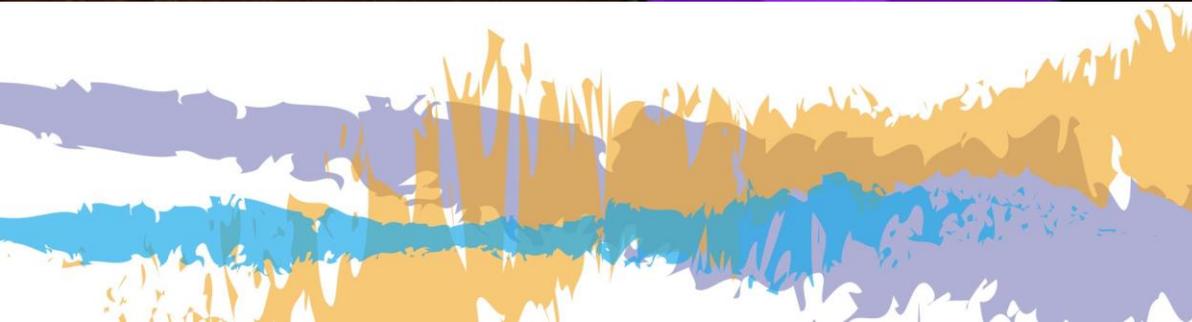
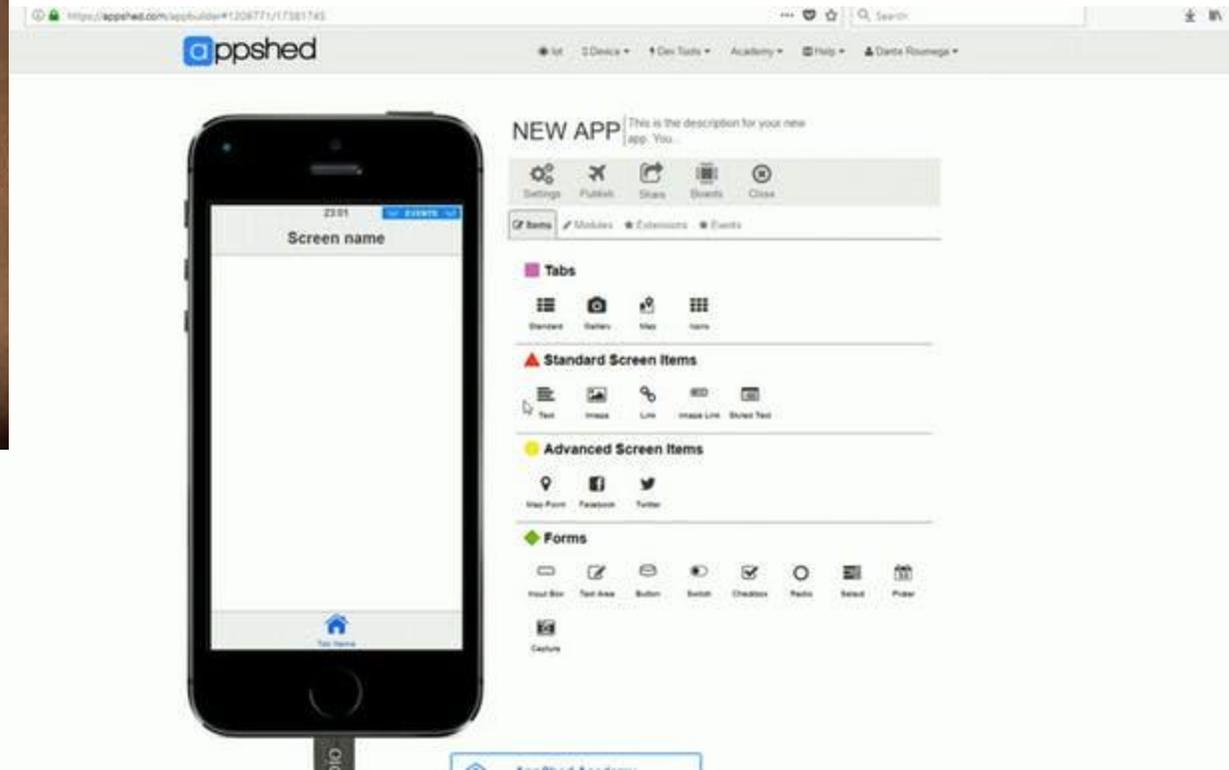
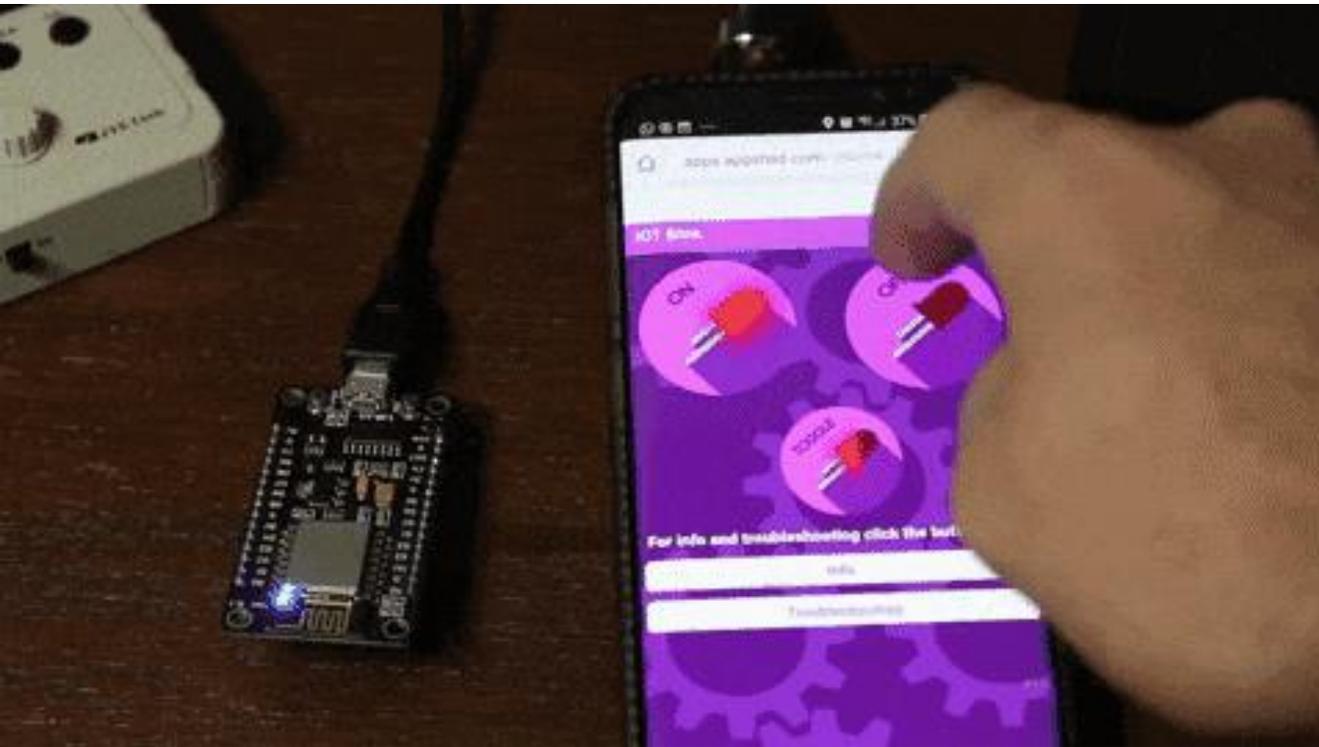




TENCODE TECHNOLOGY

#天閣科技 十行程式碼改變你對程式的想像

# 以Node MCU ESP8266 達成Blink





TENCODE TECHNOLOGY

#天閣科技 十行程式碼改變你對程式的想像

# 以Tactigon控制的Arduino動力機器人

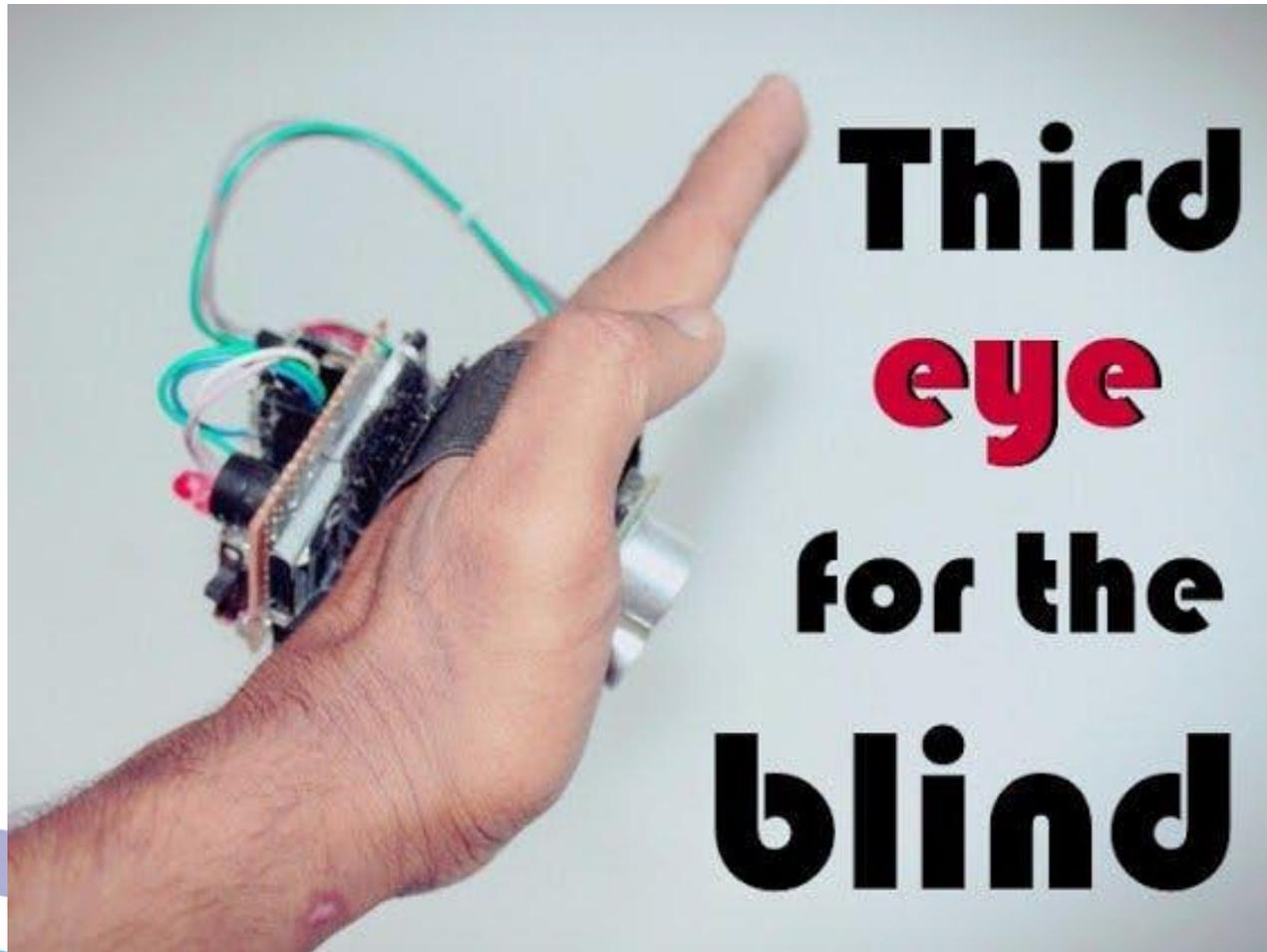




TENCODE TECHNOLOGY

#天開科技 十行程式碼改變你對程式的想像

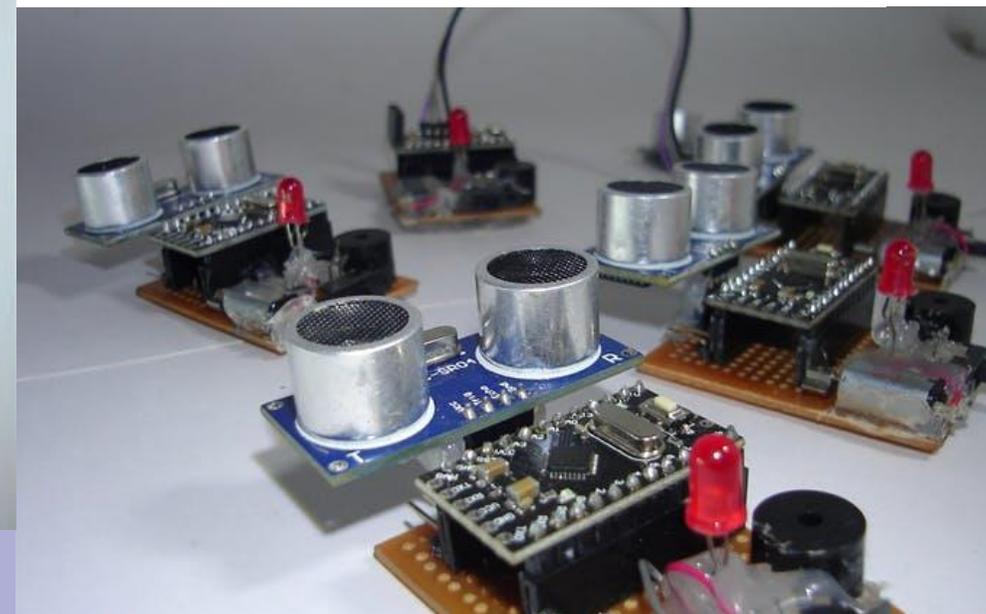
# 視障人士穿戴裝置



Position of one module

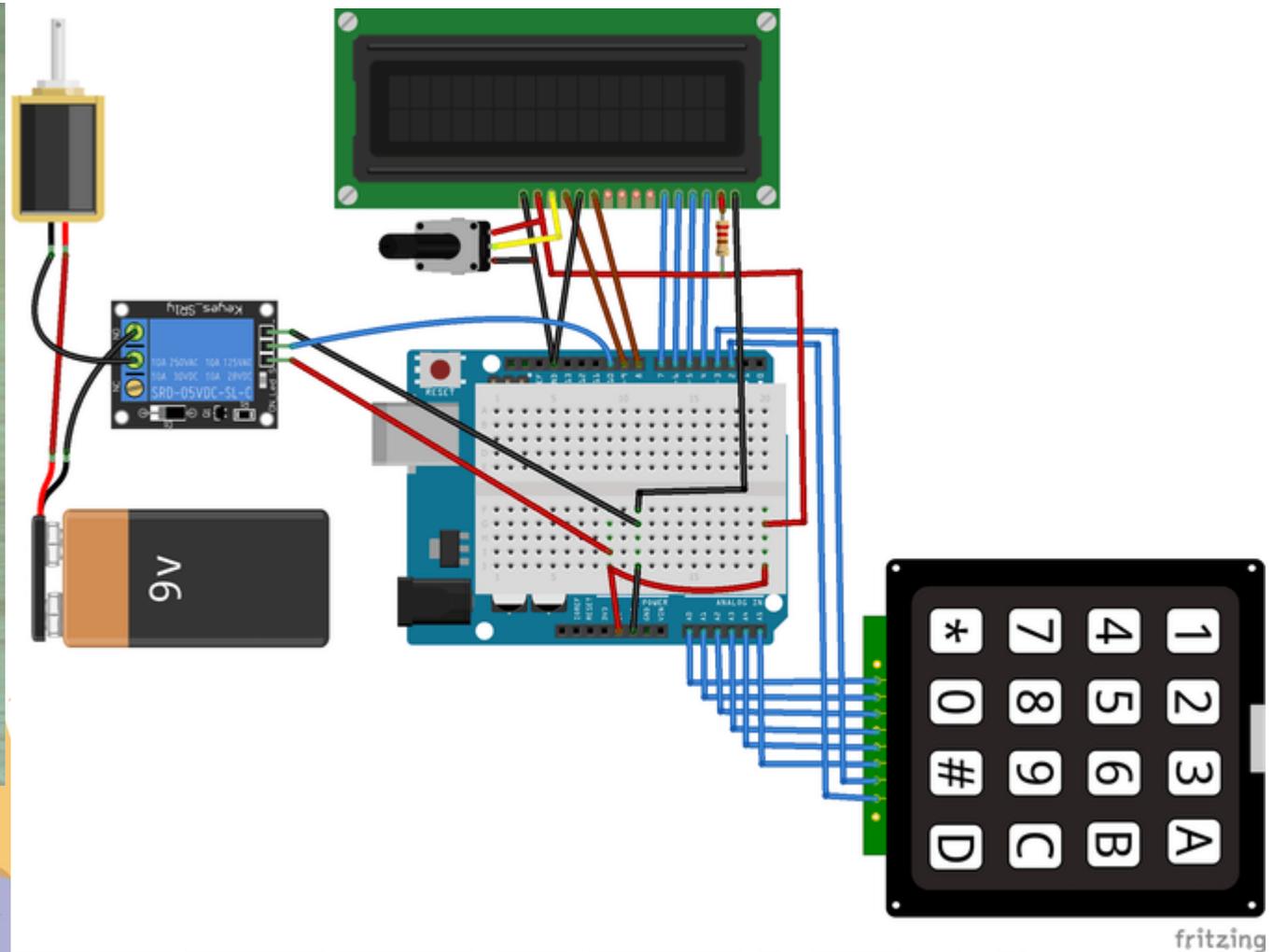


Area of detection





# 鍵盤和LCD的Arduino無鑰匙門鎖系統

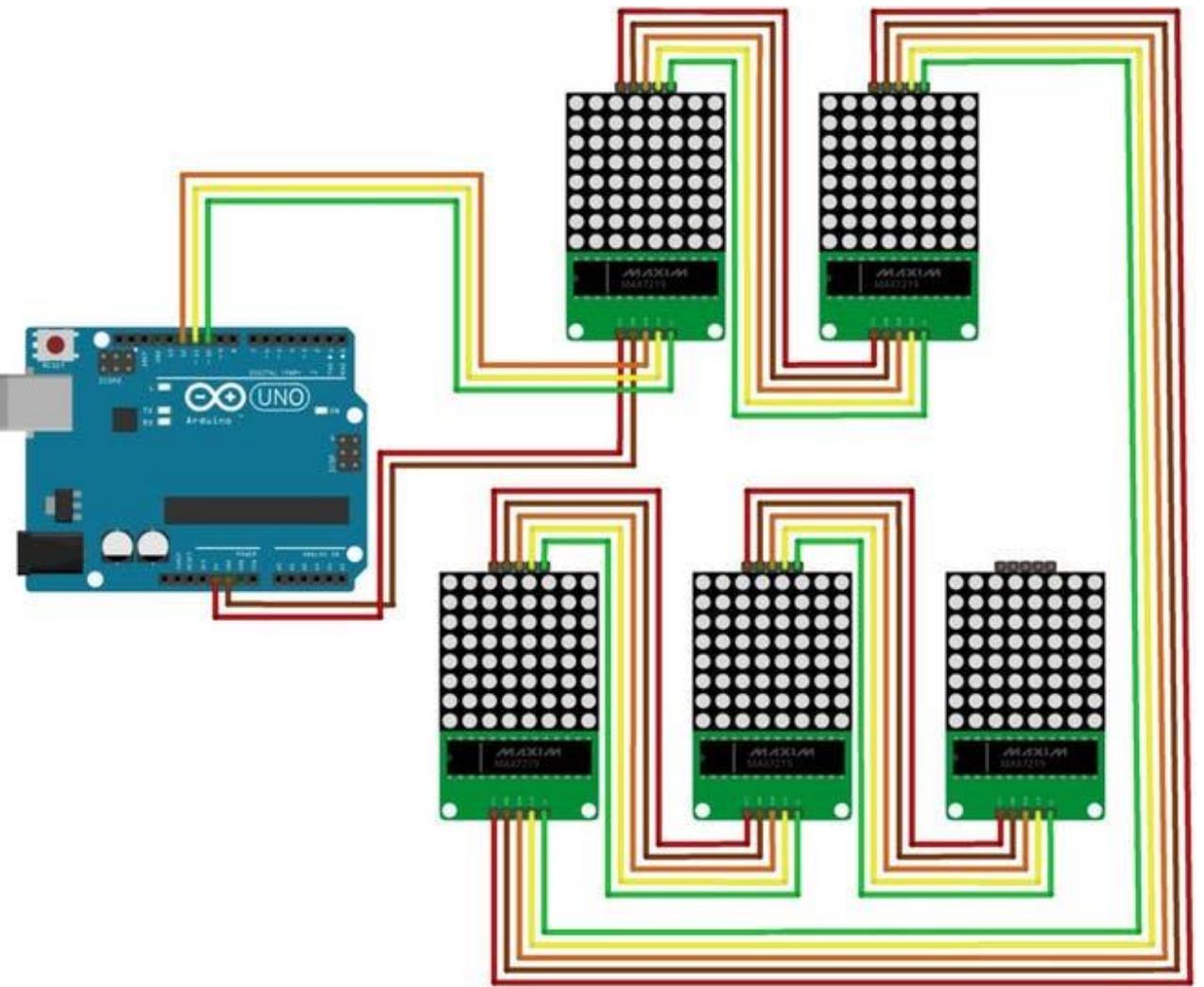
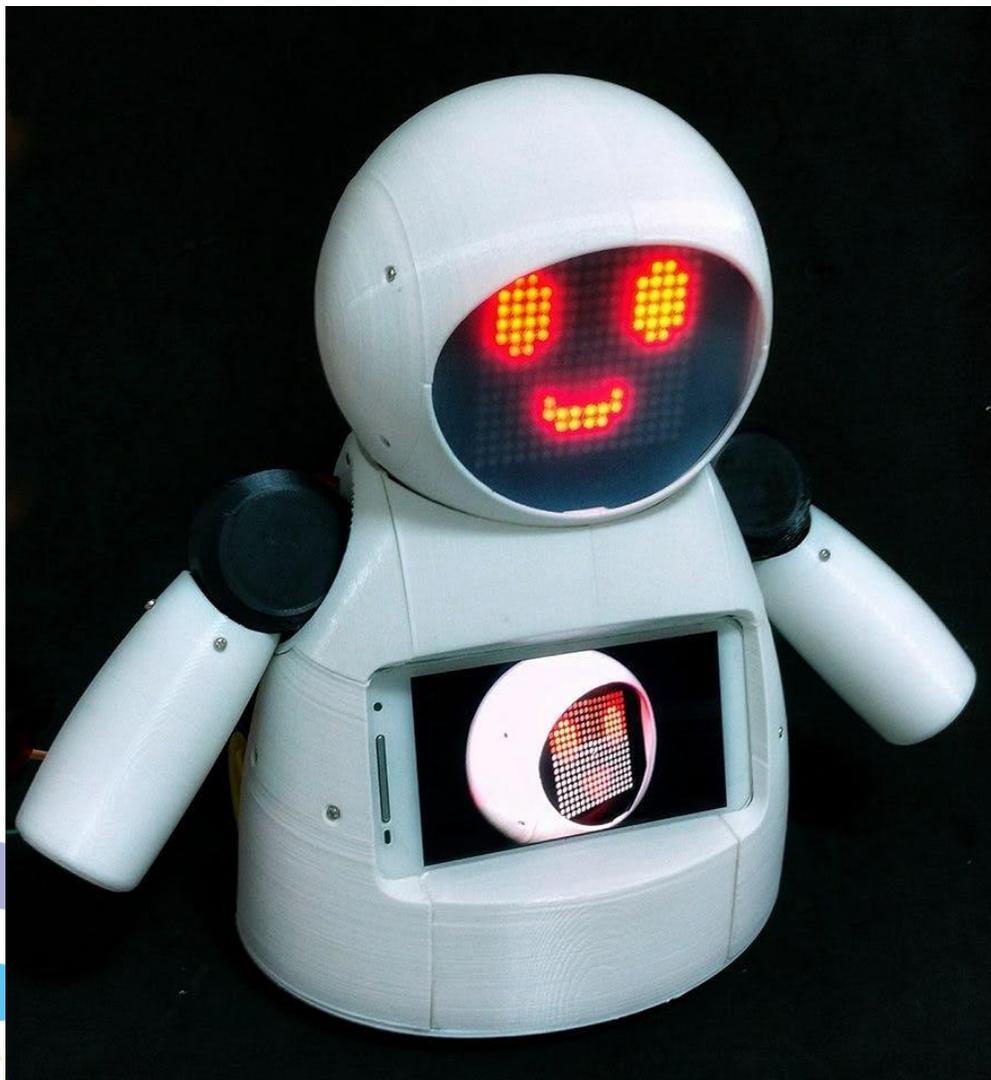




TENCODE TECHNOLOGY

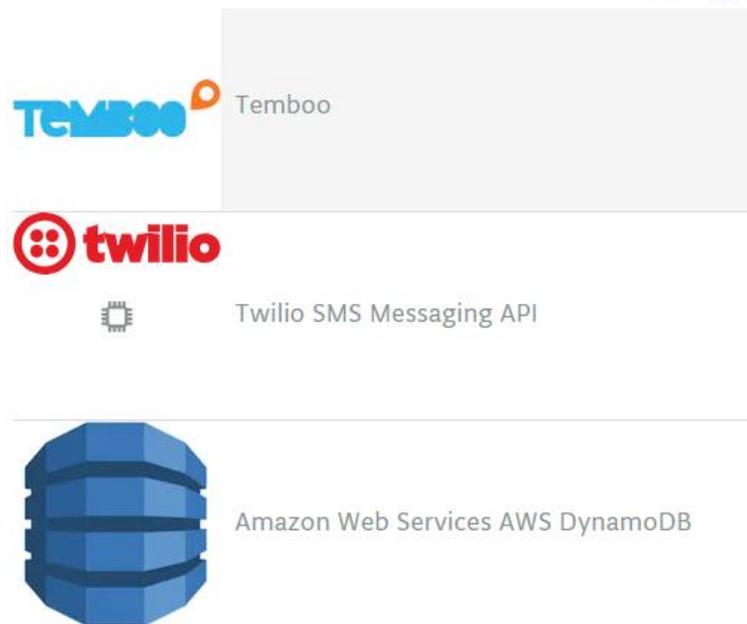
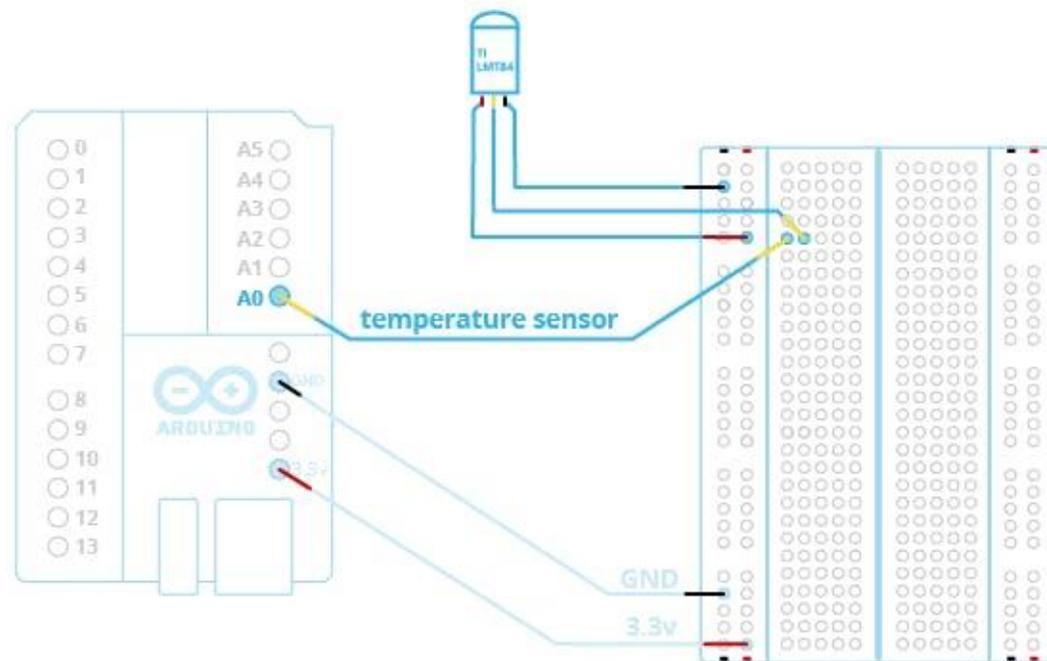
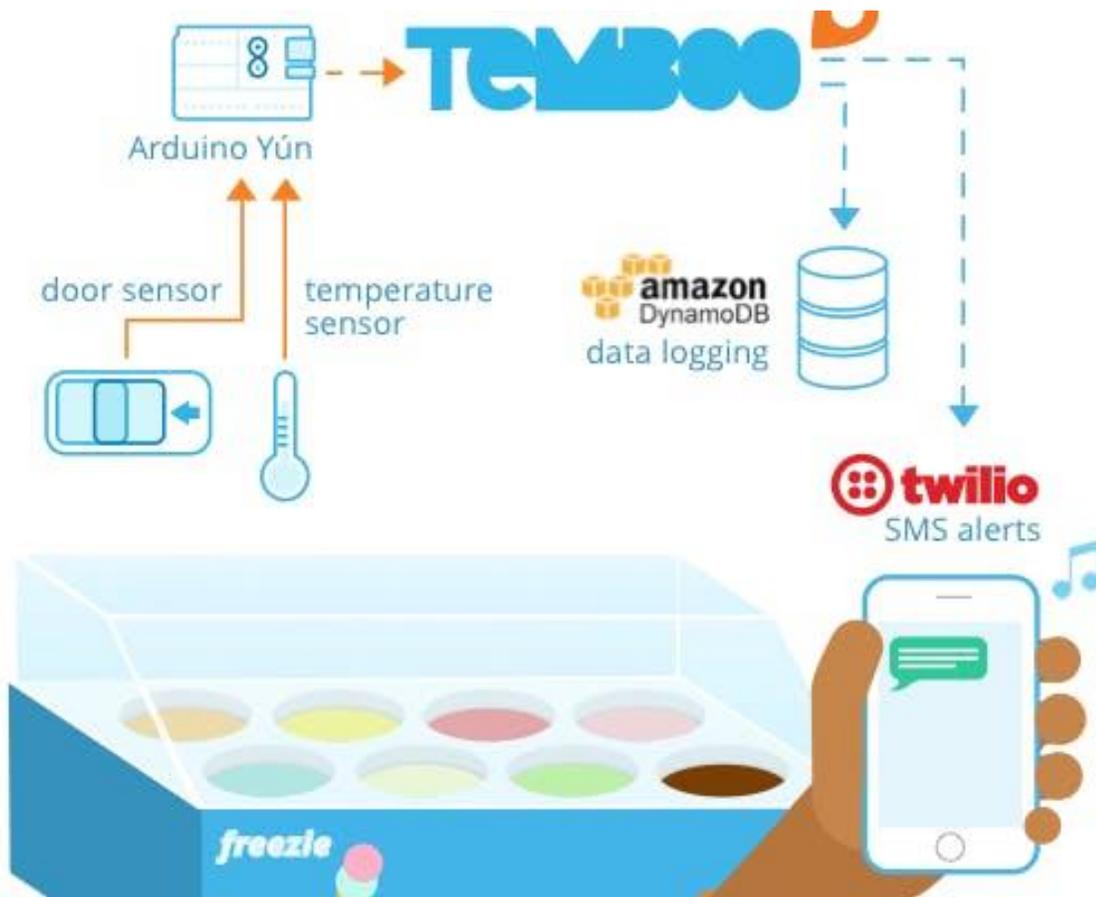
#天閣科技 十行程式碼改變你對程式的想像

# 以UNO及ESP8266達成醫院照顧兒童機器人



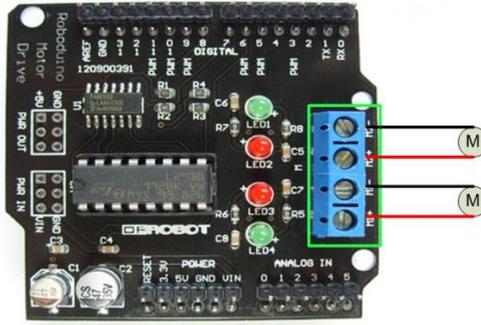


# 智能商用冷凍監視器



# Arduino感測器介紹

# Arduino與感測器產生價值



Motor Shield



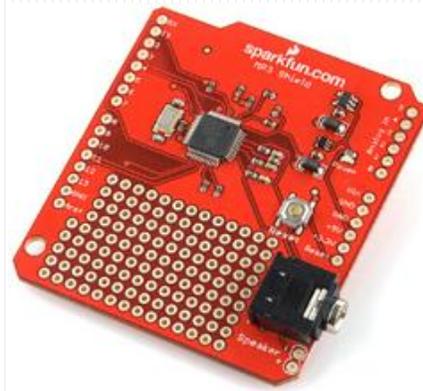
microSD Shield



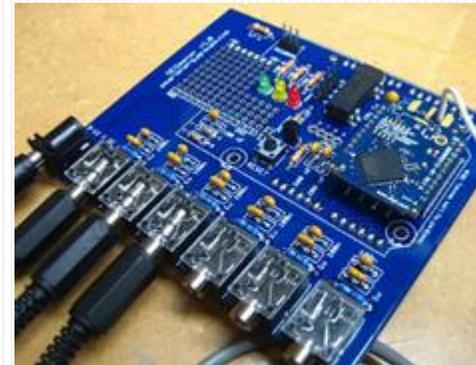
Touch Shield



Input Shield

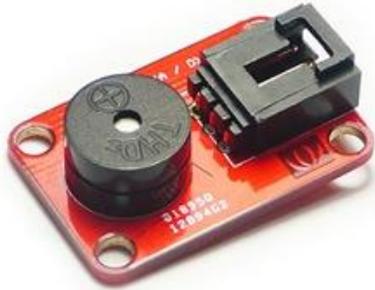


Mp3 Shield

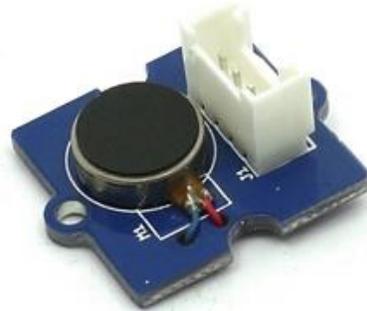


SEmeter Shield  
(Smart Meter)

# Arduino與感測器產生價值



Buzzer



Vibrator



Relay



LCD

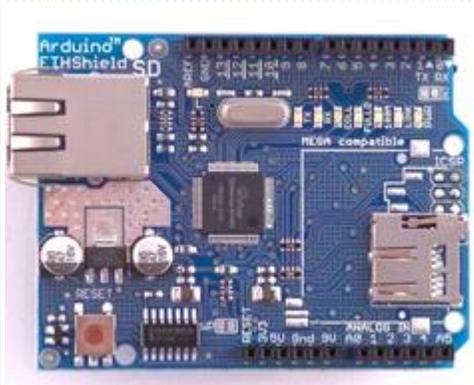


OLED

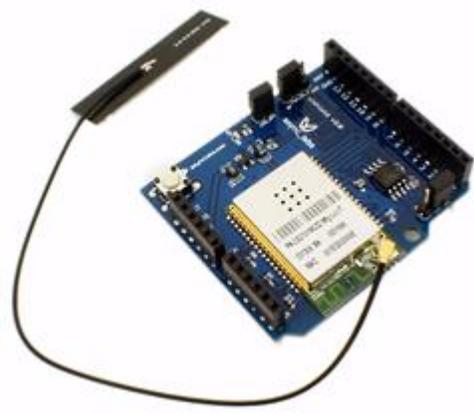


LED Matrix

# Arduino與感測器產生價值



Ethernet



WiFi



RFID



Bluetooth

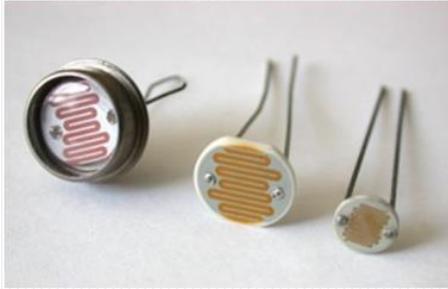


GPRS



WiFiFly

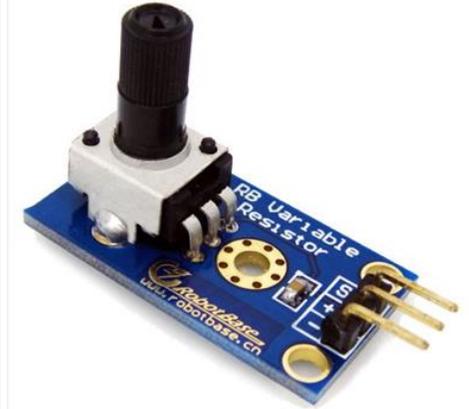
# Arduino與感測器產生價值



Photocell (light sensor)



Color sensor



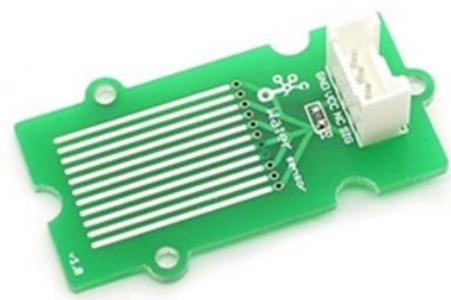
Rotation sensor (potentiometer)



Joystick

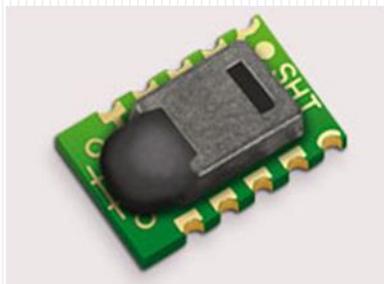


Sound Sensor



Water Sensor

# Arduino與感測器產生價值



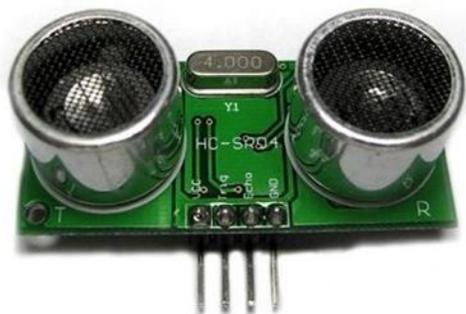
Temperature/Humidity



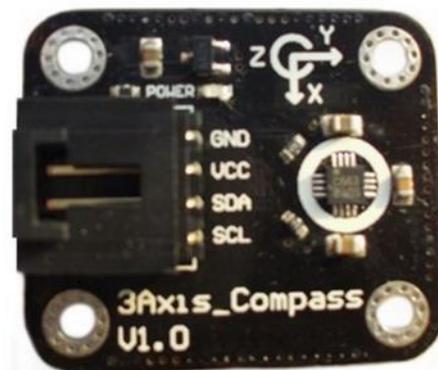
Sharp IR Sensor



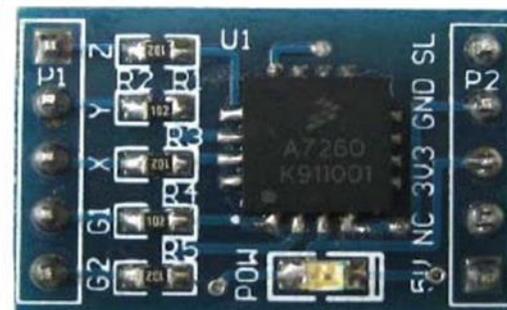
PIR



Ultrasonic



Compass (電子羅盤)

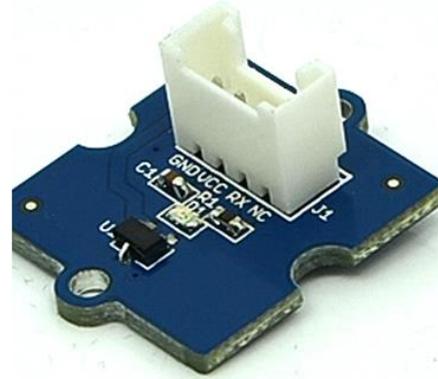


Tri-Axis Accelerometer  
(三軸加速度計)

# Arduino與感測器產生價值



AC Current Sensor



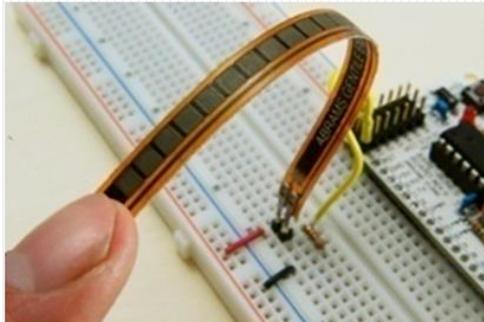
Hall Sensor (磁場感應)



CO2 Sensor



Infrared Receiver



Flex Sensor (彎曲傳感)



Flame Sensor (火焰感測)

# Arduino與感測器產生價值



IR Thermometer



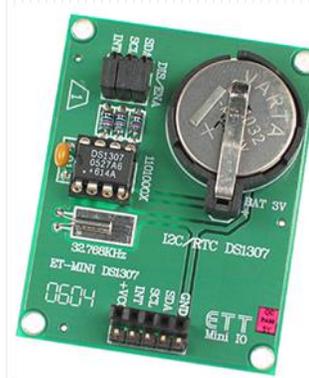
Smoke Detector



Pressure



Polar Heart Rate Sensor

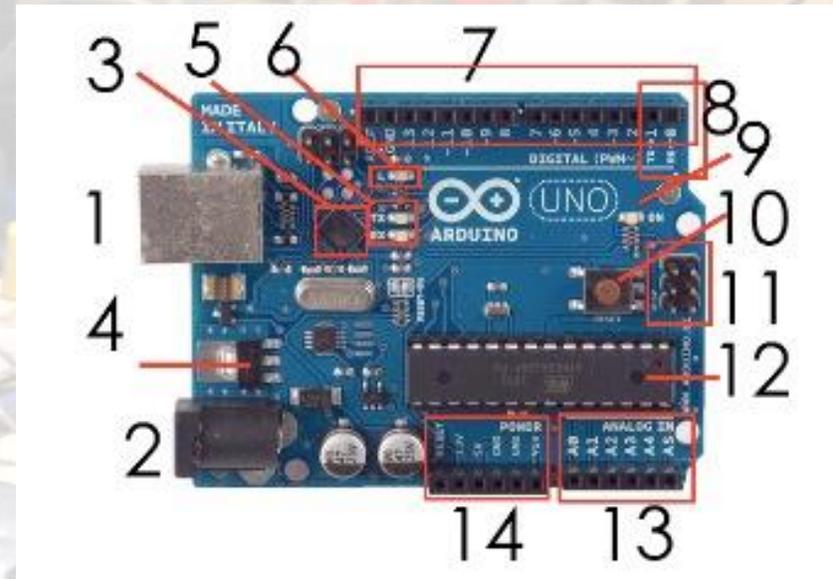


RTC



pH Sensor

1. 支援**USB接頭**傳輸資料及供電(不需額外電源)。
2. 輸入電壓Power Jack，可以單獨使用，不需連接USB電腦時就可以直接接上**5V變壓器**，或是電池直接供電。(所以1和2只要選選擇一個就可以了)
3. FTDI USB 晶片，這是USB的Client 端的晶片，透過這個晶片就可以跟**電腦溝通連結傳遞資料**。
4. Voltage Regulator:**穩壓器**，保持電壓的穩定。
5. LED 燈：用來顯示**傳遞接受的資料RX和傳遞出去的資料TX**，如有資料進去跟出去的話這LED燈都會閃爍一下，並且這兩個LED燈，和**區域7的接腳0和接腳1**是相連結的。
6. LED 燈：用來顯示**區域7的接腳13**，是高電壓還低電壓。
- 7. 數位接腳:**由右到左分別是
  - 7-1：數位接腳0到數位接腳13：需注意看一下印刷，有些字前面有~符號，例如說~10，則可將它可以當成數位接腳10，可做PWM 的資料輸出，故有此符號的意思為只有特別的接腳，可以做PWM 的資料輸出。
  - 7-2：GND: 接地接腳。
  - 7-3：AREF：AREF 是指模擬參考。全名是 Analogue REference，它可讓我們知道Arduino的參考電壓。例如，如我們要測量的最大電壓範圍3.3V，那我們就需要有一個很好的3.3V，那我們就可以從AREF接腳取得。



8.RX<- 0 和 TX -> 1:意思是說如果要做UART 資料傳遞的話，想要把資料傳進去arduino 的話，可以透過TX -> 1 (接腳1)，那arduino 如果要做做UART 資料傳遞出去的話，就可以把接腳RX<- 0 (接腳0) 接到另外一個硬體上面。此要提醒一下這兩個接腳，平常時會跟USB 連接線連結在一起了，所以說如果要把程式燒錄透過USB到這個板子上時，那個時刻請確定你的接腳0和接腳1上面是沒有任何接線的，不然程式燒錄的動作就會失敗喔！

9.LED 燈：電源顯示燈。

10.重新執行程式按鈕。

11.ICSP Header:英文的全名是In-circuit serial programming序列燒錄方式，PIC燒錄器燒錄程式的方法，程式記憶體為Flash 的版本使用方便的ICSP 序列燒錄方式。

12.Microcontroller: 控制晶片。

13.類比訊號接腳：a0,a1,a2,a3,a4,a5 **10位元解析度**

14.電源控制的接腳：

14-1: Vin (電源輸入)

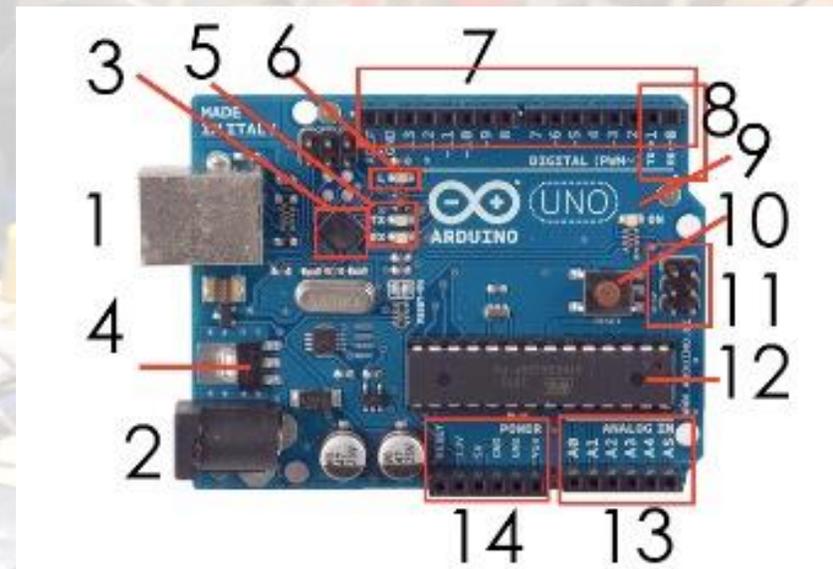
14-2: Gnd (接地)

14-3: Gnd (接地)

14-3: 5V (5V 電源輸出)

14-3: 3.3V (3.3V電源輸出)

14-3: RESET (重新執行程式)



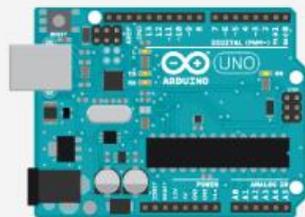


TENCODE TECHNOLOGY

# 第三堂課程

## Arduino 下載

ARDUINO是什麼？



購買ARDUINO



學習ARDUINO



支持ARDUINO



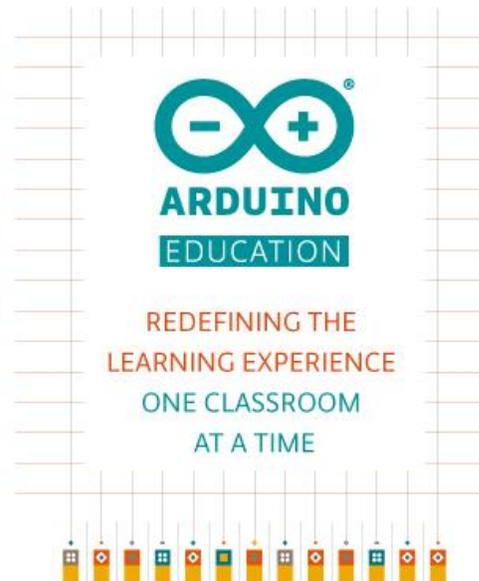
編寫代碼，製作IoT項目，並訪問酷教程！



BLOG

使用SLIDEBAR增強鍵盤

ARDUINO MKRZERO，  
零的力量  
在一個小的形式因素！  
現在去購物



BLOG

用指紋打開車庫門

# 下载Arduino软件



## ARDUINO 1.8.2

开源的Arduino软件（IDE）让编程和下载程序变得非常简单。这个软件能够运行在Windows、Mac OS X以及Linux上，软件基于Processing和其他的开源软件，使用java开发完成。

这个软件适用于任何Arduino控制板  
具体的使用说明可以参照 [快速入门](#) 的页面

**Windows** 安装包

**Windows** 免安装ZIP包

**Windows** 移动



**Mac OS X** 10.7 Lion 或更新的版本

**Linux** 32 位

**Linux** 64 位

**Linux** ARM 位

版本说明

源代码

校验

## Arduino Setup: License Agreement



Please review the license agreement before installing Arduino. If you accept all terms of the agreement, click I Agree.

### GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

Cancel

Nullsoft Install System v3.0

I Agree



## Arduino Setup: Installation Options



Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

- Install Arduino software
- Install USB driver
- Create Start Menu shortcut
- Create Desktop shortcut
- Associate .ino files

Space required: 418.6MB

Cancel

Nullsoft Install System v3.0

< Back

Next >



Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder

C:\Program Files (x86)\Arduino\

Browse...

Space required: 418.6MB

Space available: 5.1GB

Cancel

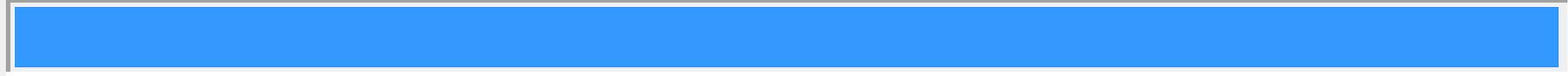
Nullsoft Install System v3.0

< Back

Install



Completed



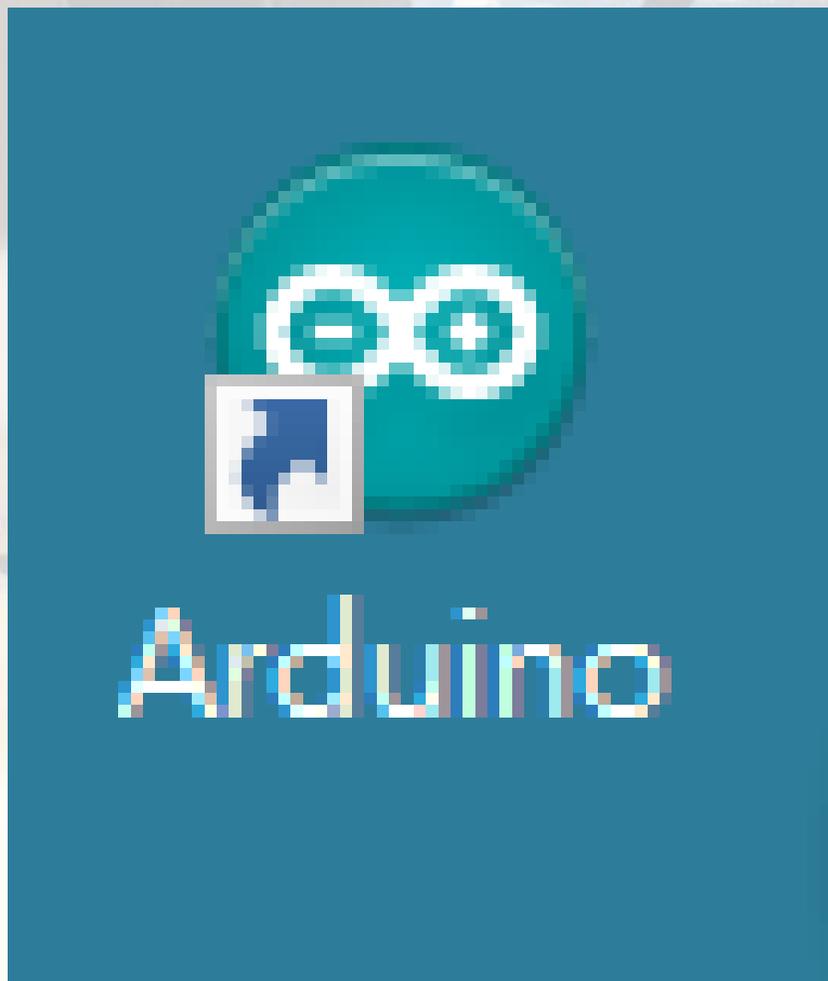
Show details

Cancel

Nullsoft Install System v3.0

< Back

Close

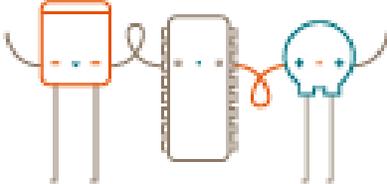


ARDUINO

Genuino

AN OPEN PROJECT WRITTEN, DEBUGGED,  
AND SUPPORTED BY ARDUINO.CC AND  
THE ARDUINO COMMUNITY WORLDWIDE

LEARN MORE ABOUT THE CONTRIBUTORS  
OF **ARDUINO.CC** on [arduino.cc/credits](https://arduino.cc/credits)



正在準備開發板



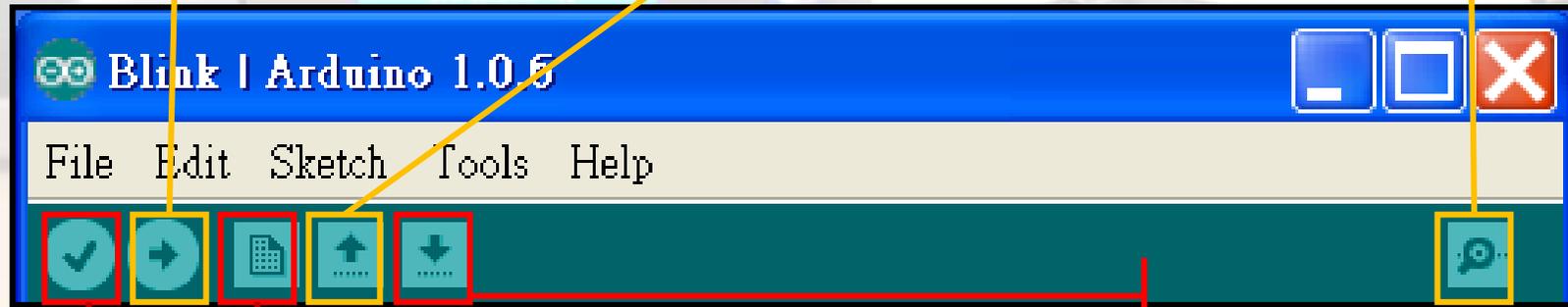
sketch\_apr29a

```
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

將程式燒入到  
Arduino

開啟舊檔

回傳值



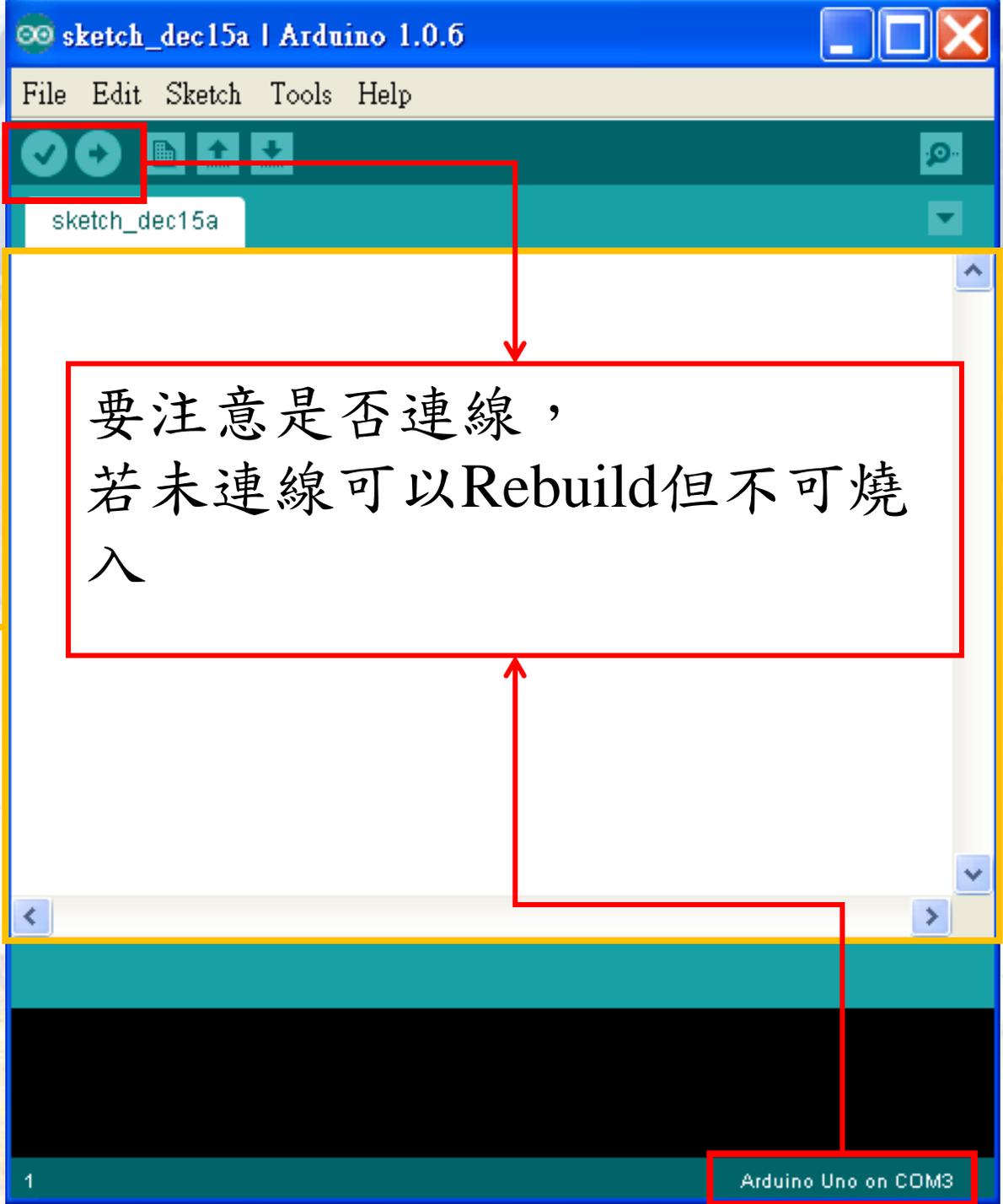
同等於C語言的  
Rebuild All

開新檔案

儲存檔案

撰寫程式內容

要注意是否連線，  
若未連線可以Rebuild但不可燒  
入





# Arduino 程式架構

## (一)宣告:

Arduino 宣告方式與 C 相同,例如:int , float...

## (二)初始化 Setup():

使 Arduino 板子裝置妥當的指令

### EX:

```
int ledPin=7;    /*宣告Arduino7號腳為輸入腳*/
```

```
  Setup()
```

```
{
```

```
    pinMode(ledPin,INPUT);
```

```
}
```



### (三)執行Loop():

為程式的主要內容,這程式內容會一直重複被執行

EX:

```
Loop()
```

```
{
```

```
.....
```

```
}
```



#### (四)函式:

1. pinMode(7,INPUT)

//將腳位7設定為輸入模式

2. digitalWrite(8,HIGH) (數位腳專用)

//將腳位8設定輸出高電位

3. val=digitalRead(7) (數位腳專用)

//讀出腳位7的值並指定給

val變數

4. `analogWrite(9,128)` (數位訊號專用所設計的函式)

//將擁有PWM的數位腳位9  
設定輸出電位2.5V對應值大約為128

5. `val=analogRead(0)` (類比腳專用)

//讀出腳位0的值並指定給val變數(且`analogRead`可讀取  
範圍0(0V)~1023(5V))

PowerPoint window: .pptx - PowerPoint

Arduino IDE window: AnalogReadSerial | Arduino 1.8.2

Arduino IDE menu: 開發板管理員...

- Arduino AVR 板
- Arduino Yún
- **Arduino/Genuino Uno**
- Arduino Duemilanove or Diecimila
- Arduino Nano
- Arduino/Genuino Mega or Mega 2560
- Arduino Mega ADK
- Arduino Leonardo
- Arduino Leonardo ETH
- Arduino/Genuino Micro
- Arduino Esplora
- Arduino Mini
- Arduino Ethernet
- Arduino Fio
- Arduino BT
- LilyPad Arduino USB
- LilyPad Arduino
- Arduino Pro or Pro Mini
- Arduino NG or older
- Arduino Robot Control
- Arduino Robot Motor
- Arduino Gemma
- Adafruit Circuit Playground
- Arduino Yún Mini
- Arduino Industrial 101
- Linino One
- Arduino Uno WiFi

Arduino IDE toolbar: 自動格式化 (Ctrl+T), 封存草稿碼, 修正編碼並重新載入, 序列埠監控視窗 (Ctrl+Shift+M), 序列繪圖家 (Ctrl+Shift+L), WiFi101 Firmware Updater, 開發板: "Arduino/Genuino Uno", 序列埠: "COM3 (Arduino/Genuino Uno)", 取得開發板資訊, 燒錄器: "AVRISP mkII", 燒錄Bootloader

```

// the setup routine
void setup() {
  // initialize serial communication:
  Serial.begin(9600);
}

// the loop routine
void loop() {
  // read the input sensor value:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1);        // delay in between reads for stability
}

```

Arduino IDE status bar: Arduino/Genuino Uno 於 COM3

Arduino IDE window: AnalogReadSerial | Arduino 1.8.2

Arduino IDE toolbar: 自動格式化 (Ctrl+T), 封存草稿碼, 修正編碼並重新載入, 序列埠監控視窗 (Ctrl+Shift+M), 序列繪圖家 (Ctrl+Shift+L), WiFi101 Firmware Updater, 開發板: "Arduino/Genuino Uno", 序列埠: "COM3 (Arduino/Genuino Uno)", 取得開發板資訊, 燒錄器: "AVRISP mkII", 燒錄Bootloader

Serial Port Selection dialog:

- 序列埠
- COM1
- COM3 (Arduino/Genuino Uno)

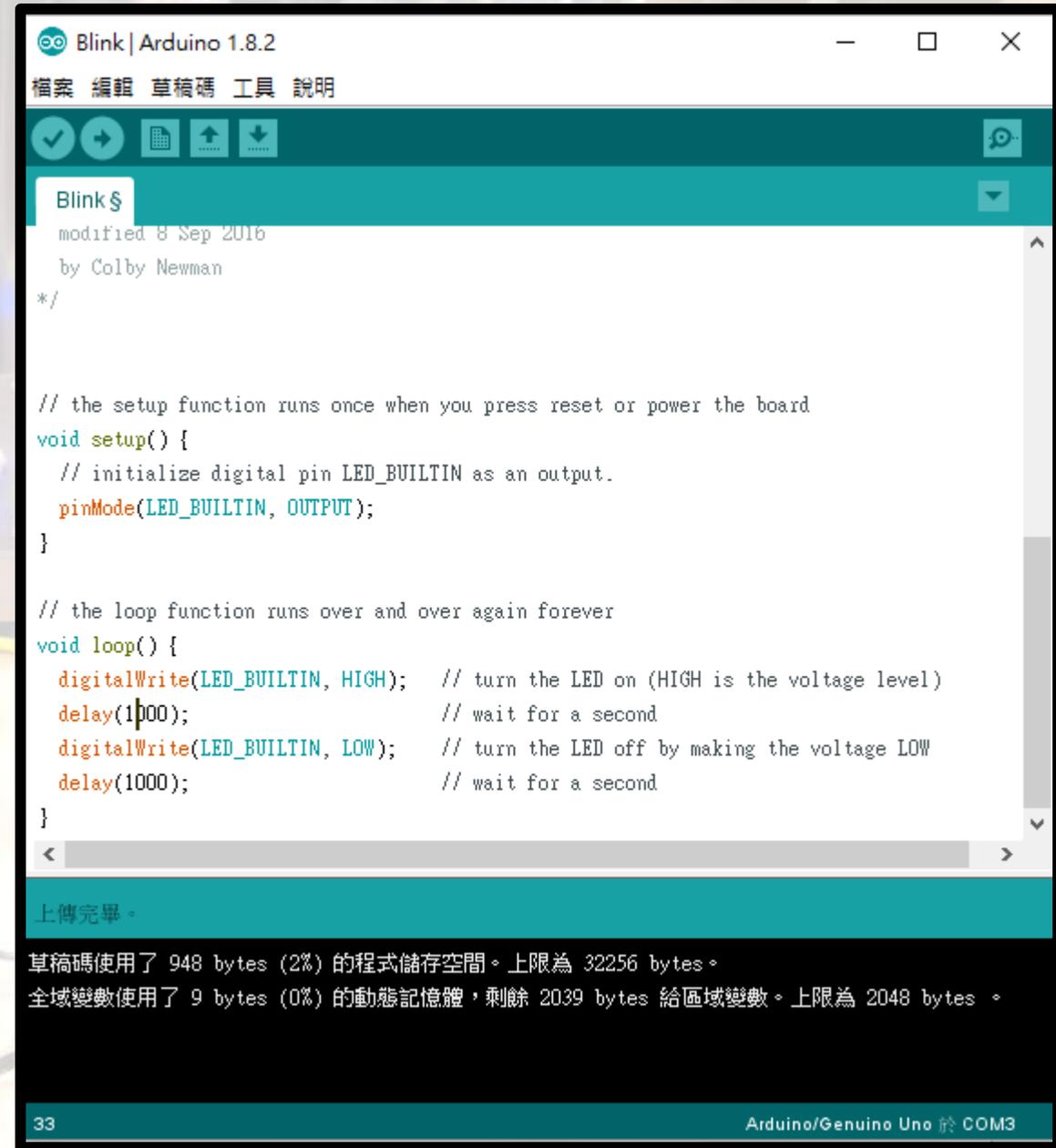
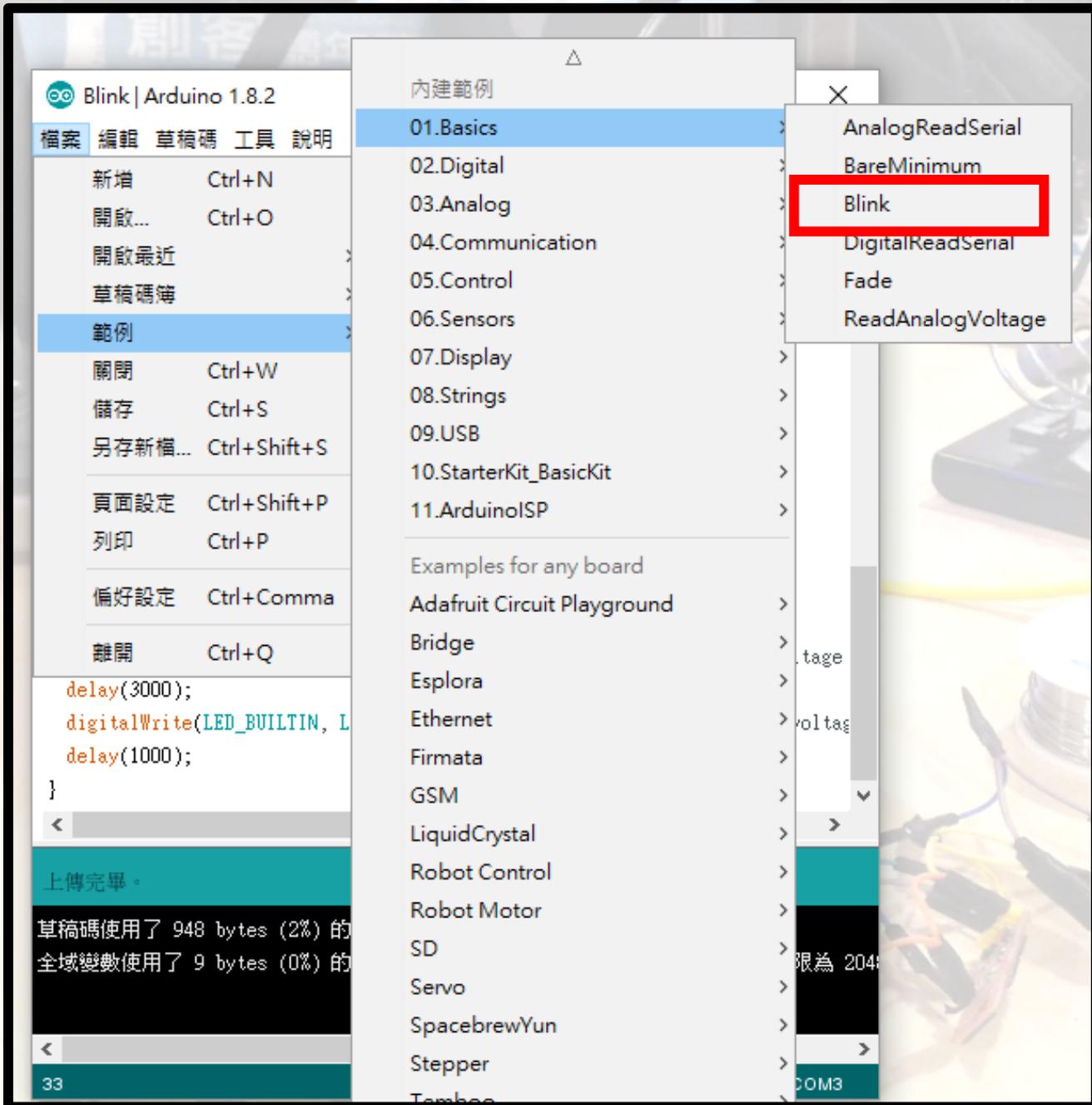
```

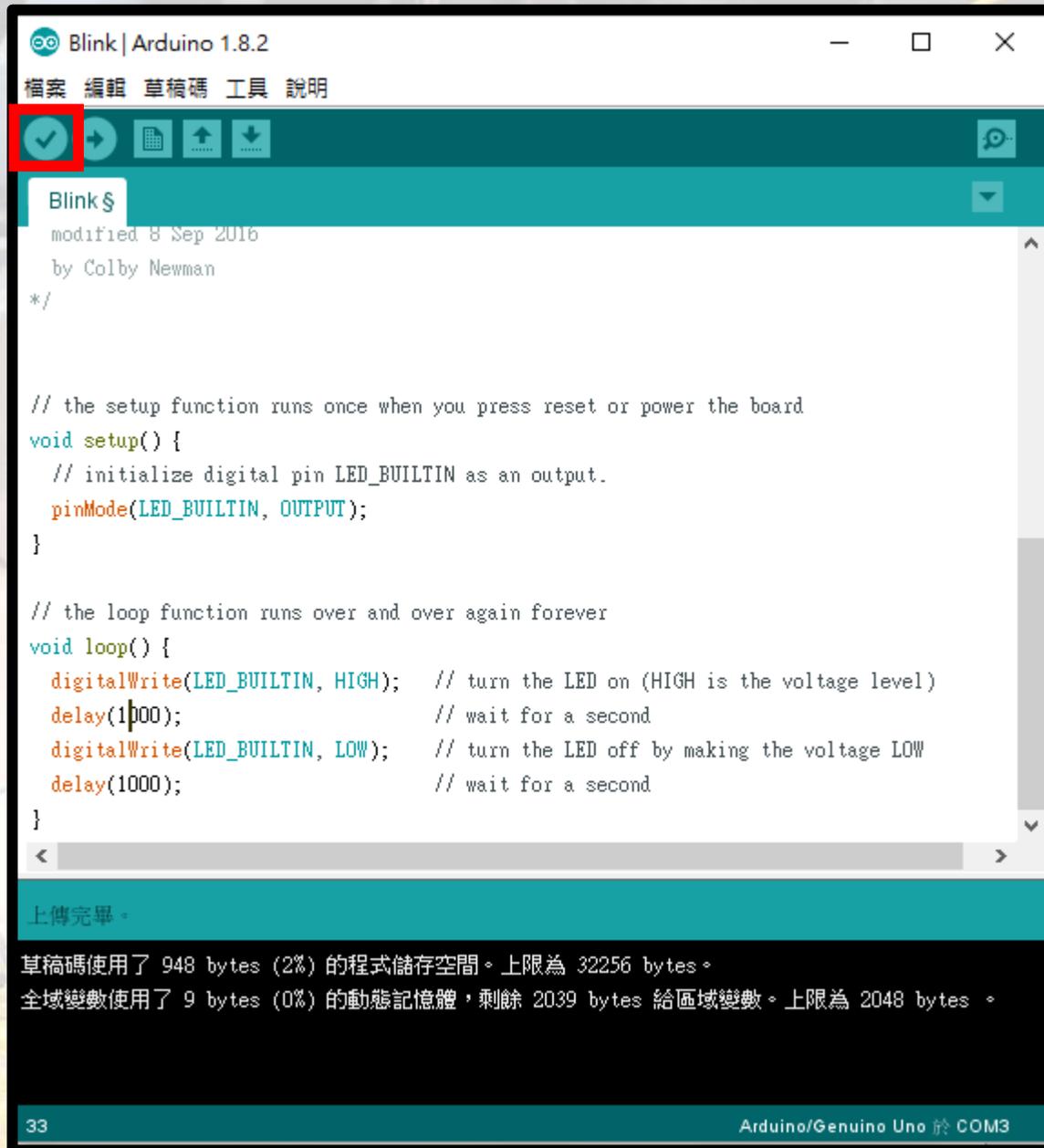
// the setup routine
void setup() {
  // initialize serial communication:
  Serial.begin(9600);
}

// the loop routine
void loop() {
  // read the input sensor value:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1);        // delay in between reads for stability
}

```

Arduino IDE status bar: Arduino/Genuino Uno 於 COM3





Blink | Arduino 1.8.2

檔案 編輯 草稿碼 工具 說明

✓ → 📄 ⬆️ ⬇️ 🔍

Blink \$

modified 8 Sep 2016  
by Colby Newman

```
*/  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```

上傳完畢。

草稿碼使用了 948 bytes (2%) 的程式儲存空間。上限為 32256 bytes。  
全域變數使用了 9 bytes (0%) 的動態記憶體，剩餘 2039 bytes 給區域變數。上限為 2048 bytes。

33 Arduino/Genuino Uno 於 COM3

Blink | Arduino 1.8.2  
檔案 編輯 草稿碼 工具 說明

✓ → ↻ ↕ ↕ 🔍

Blink \$  
modified 8 Sep 2016  
by Colby Newman  
\*/

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

上傳完畢。

草稿碼使用了 948 bytes (2%) 的程式儲存空間。上限為 32256 bytes。  
全域變數使用了 9 bytes (0%) 的動態記憶體，剩餘 2039 bytes 給區域變數。上限為 2048 bytes。

33 Arduino/Genuino Uno 於 COM3

# 編譯成功

草稿碼使用了 928 bytes (2%) 的程式儲存空間。上限為 32256 bytes。  
全域變數使用了 9 bytes (0%) 的動態記憶體，剩餘 2039 bytes 給區域變數。上限為 2048 bytes。

開發板 Arduino/Genuino Uno 編譯錯誤。

複製錯誤訊息

exit status 1  
開發板 Arduino/Genuino Uno 編譯錯誤。

31

Arduino/Genuino Uno 於 COM3

# 編譯失敗

Blink | Arduino 1.8.2

檔案 編輯 草稿碼 工具 說明

Blink \$

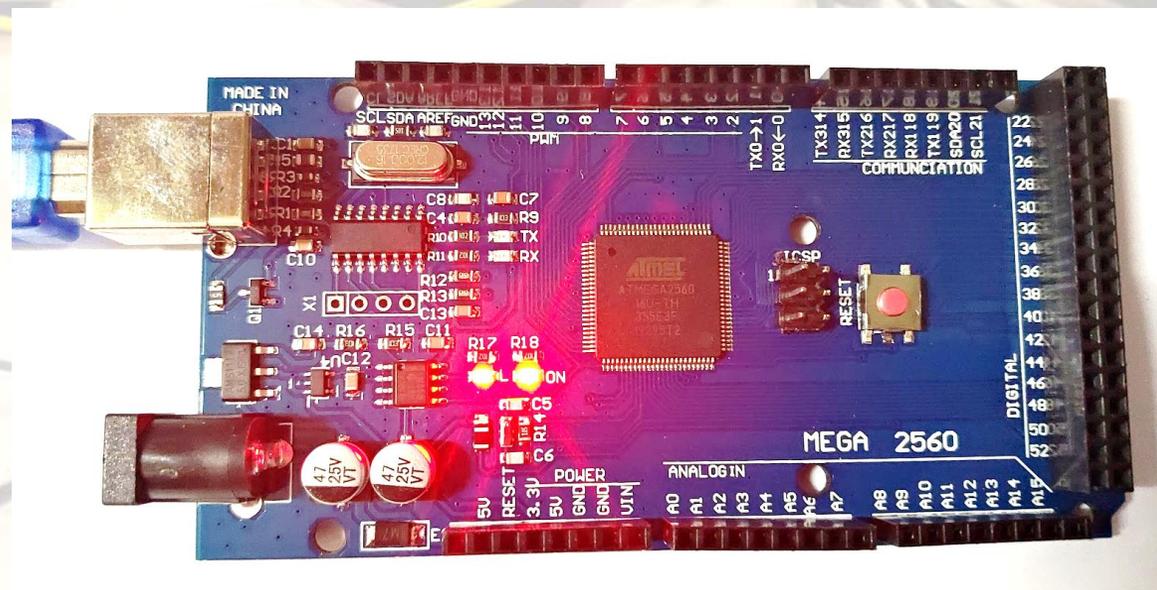
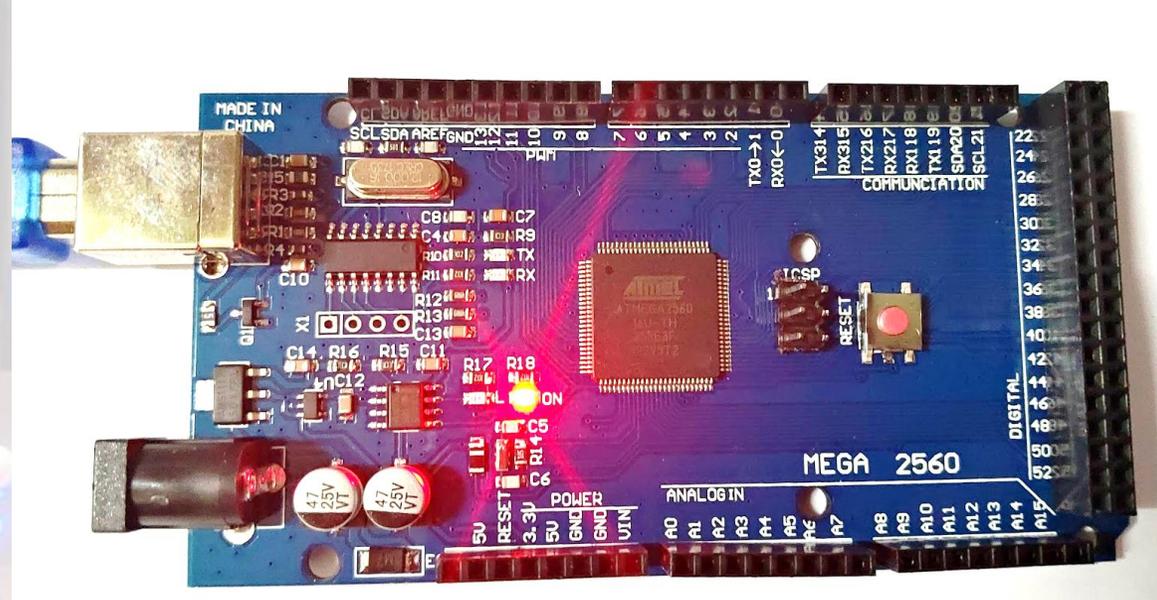
```
modified 8 Sep 2016
by Colby Newman
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

上傳完畢。
草稿碼使用了 948 bytes (2%) 的程式儲存空間。上限為 32256 bytes。
全域變數使用了 9 bytes (0%) 的動態記憶體，剩餘 2039 bytes 給區域變數。上限為 2048 bytes。
```

33 Arduino/Genuino Uno 於 COM3



```
Blink | Arduino 1.8.2
檔案 編輯 草稿碼 工具 說明
✓ → 📄 ⬆️ ⬇️ 🔍
Blink $
modified 8 Sep 2016
by Colby Newman
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

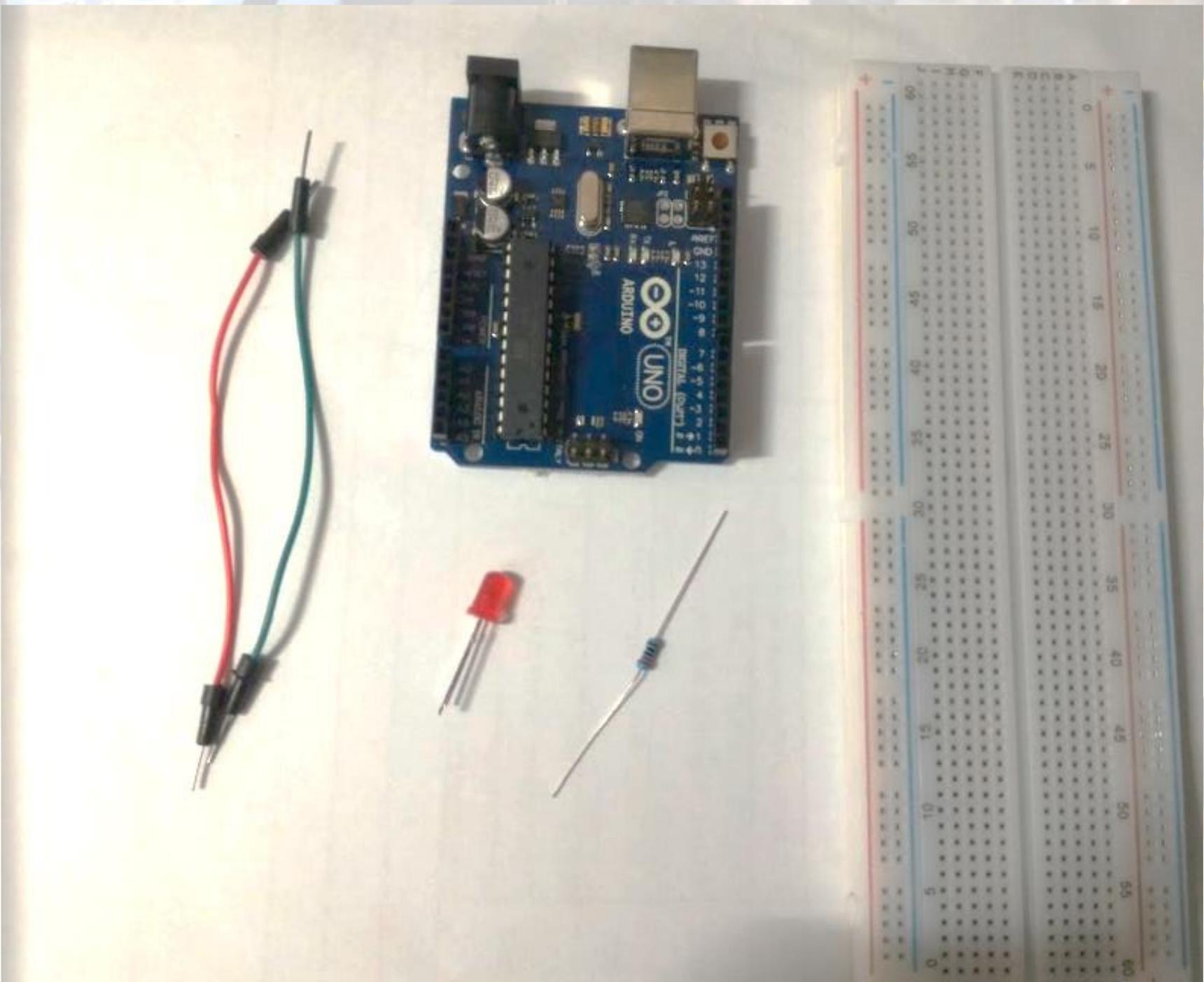
上傳完畢。
草稿碼使用了 948 bytes (2%) 的程式儲存空間。上限為 32256 bytes。
全域變數使用了 9 bytes (0%) 的動態記憶體，剩餘 2039 bytes 給區域變數。上限為 2048 bytes。
33 Arduino/Genuino Uno 於 COM3
```

```
Blink | Arduino 1.8.2
檔案 編輯 草稿碼 工具 說明
✓ → 📄 ⬆️ ⬇️ 🔍
Blink $
modified 8 Sep 2016
by Colby Newman
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

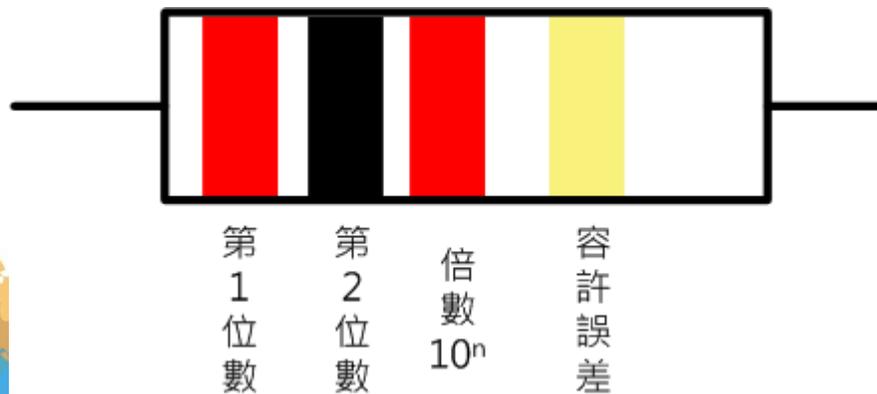
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(4000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

上傳完畢。
草稿碼使用了 928 bytes (2%) 的程式儲存空間。上限為 32256 bytes。
全域變數使用了 9 bytes (0%) 的動態記憶體，剩餘 2039 bytes 給區域變數。上限為 2048 bytes。
33 Arduino/Genuino Uno 於 COM3
```



# 一般電阻標示

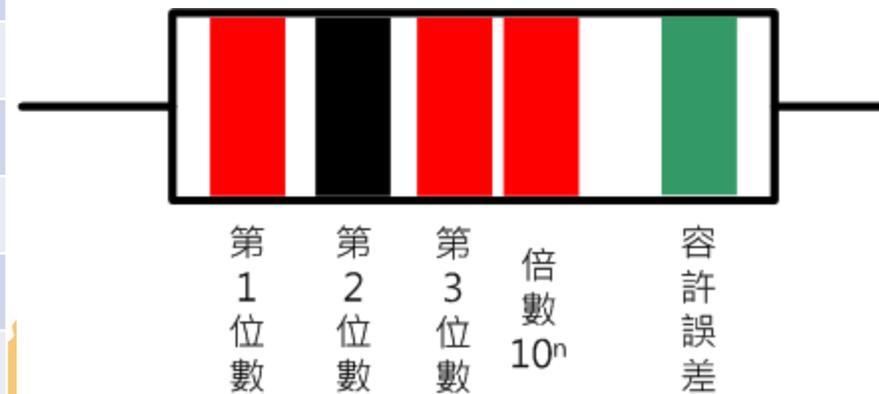
顏色	表示意義	顏色	表示意義
黑	數值0	藍	數值6
棕	數值1	紫	數值7
紅	數值2	灰	數值8
橙	數值3	白	數值9
黃	數值4	金	容許誤差±5%
綠	數值5	銀	容許誤差±10%

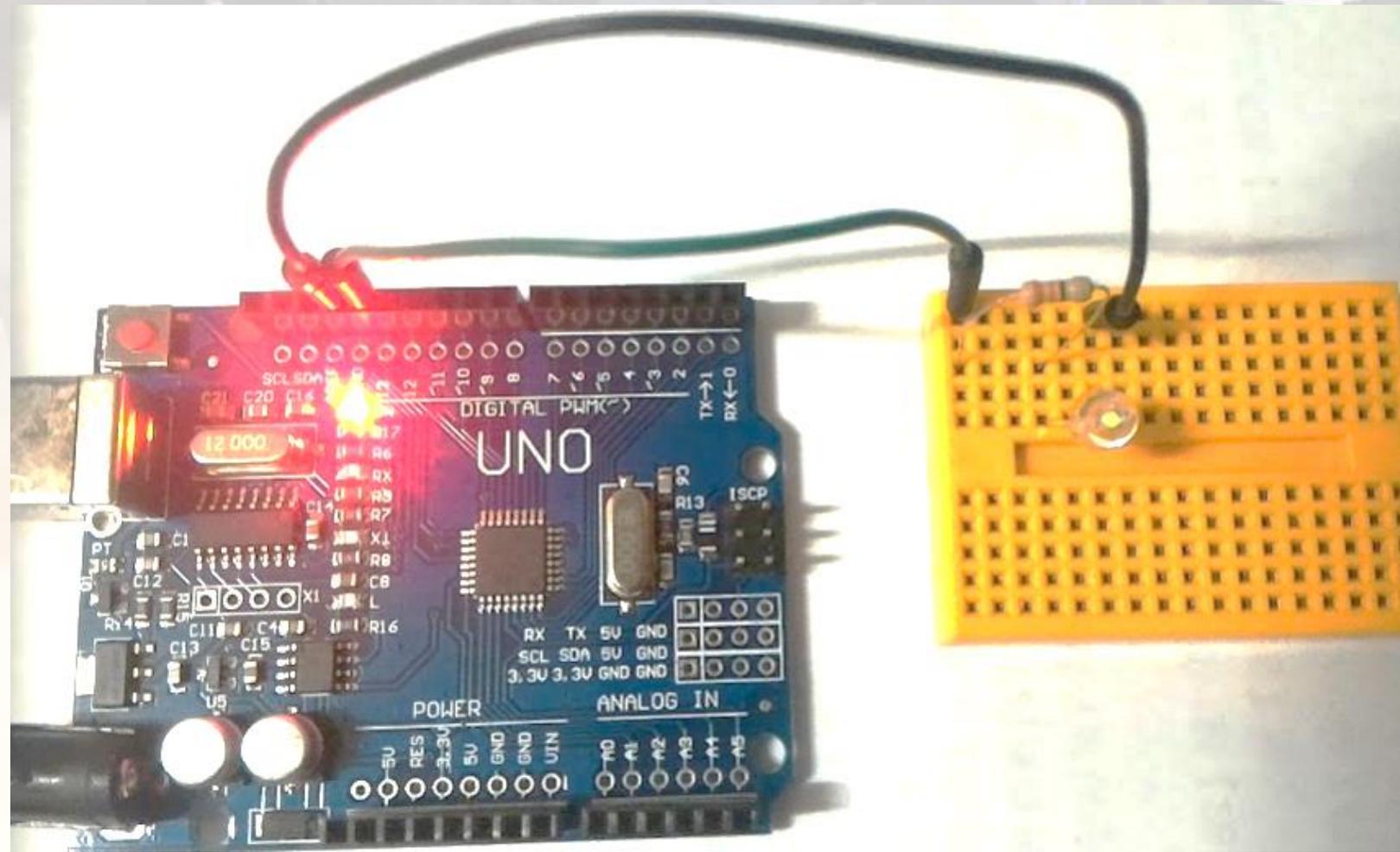


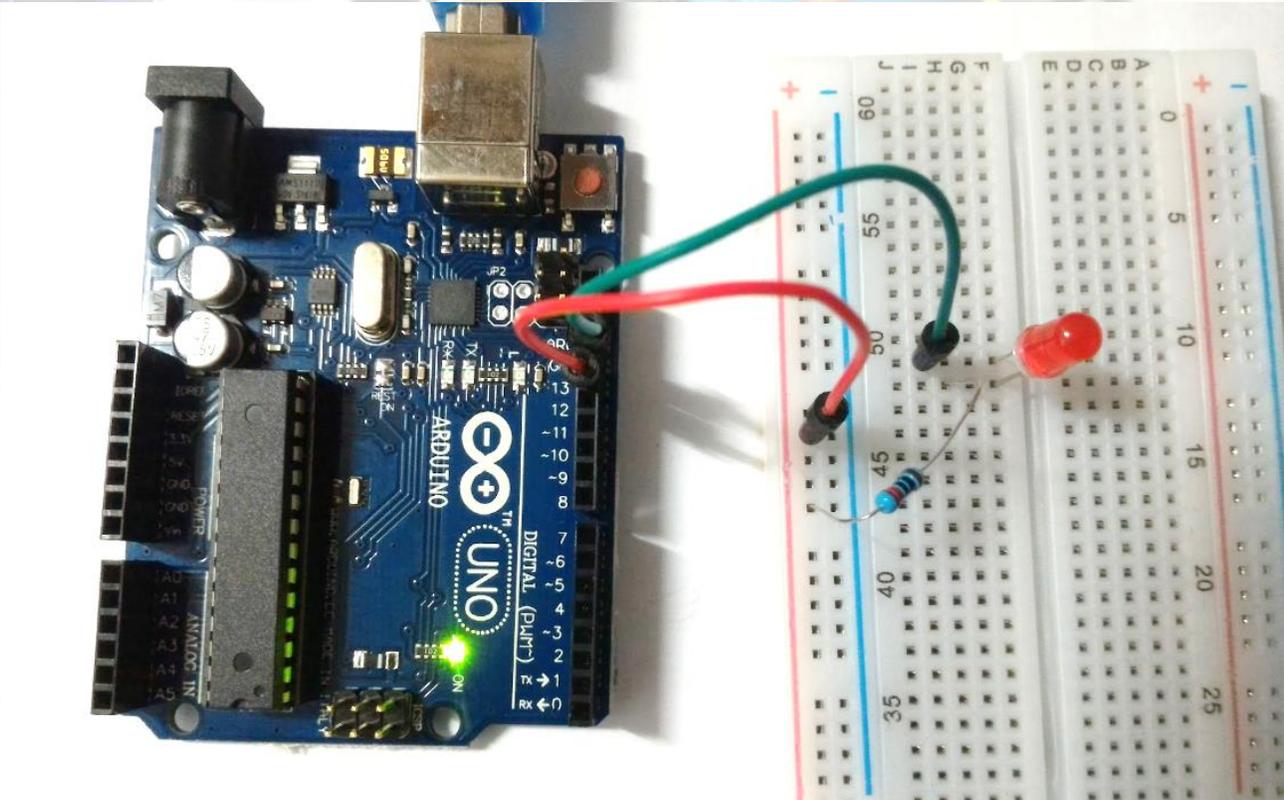
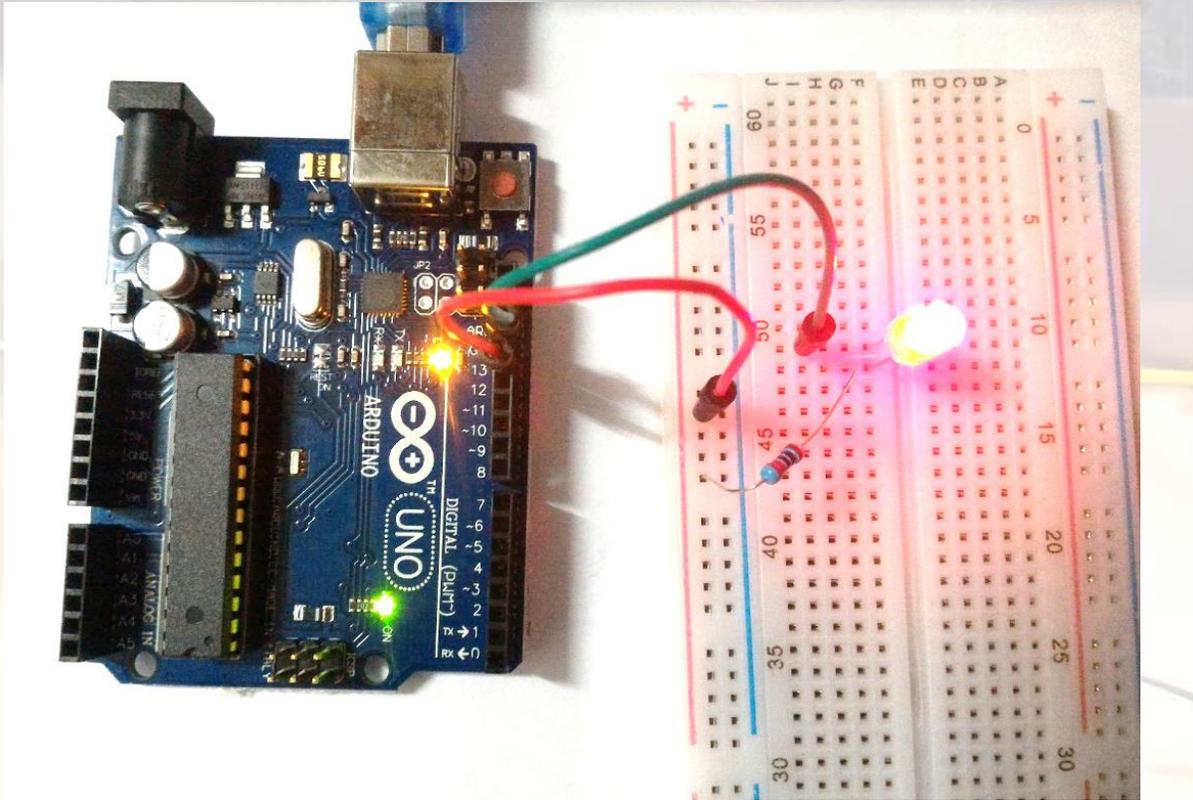


# 精密電阻標示

顏色	代表意義	代表誤差
黑	數值0	
棕	數值1	±1%
紅	數值2	±2%
橙	數值3	
黃	數值4	
綠	數值5	±0.5%
藍	數值6	±0.25%
紫	數值7	±0.1%
灰	數值8	±0.05%
白	數值9	
金		±5%
銀		±10%









TENCODE TECHNOLOGY

# 第四堂課程

## 光敏電阻的輸入

# 主題一 感光控制

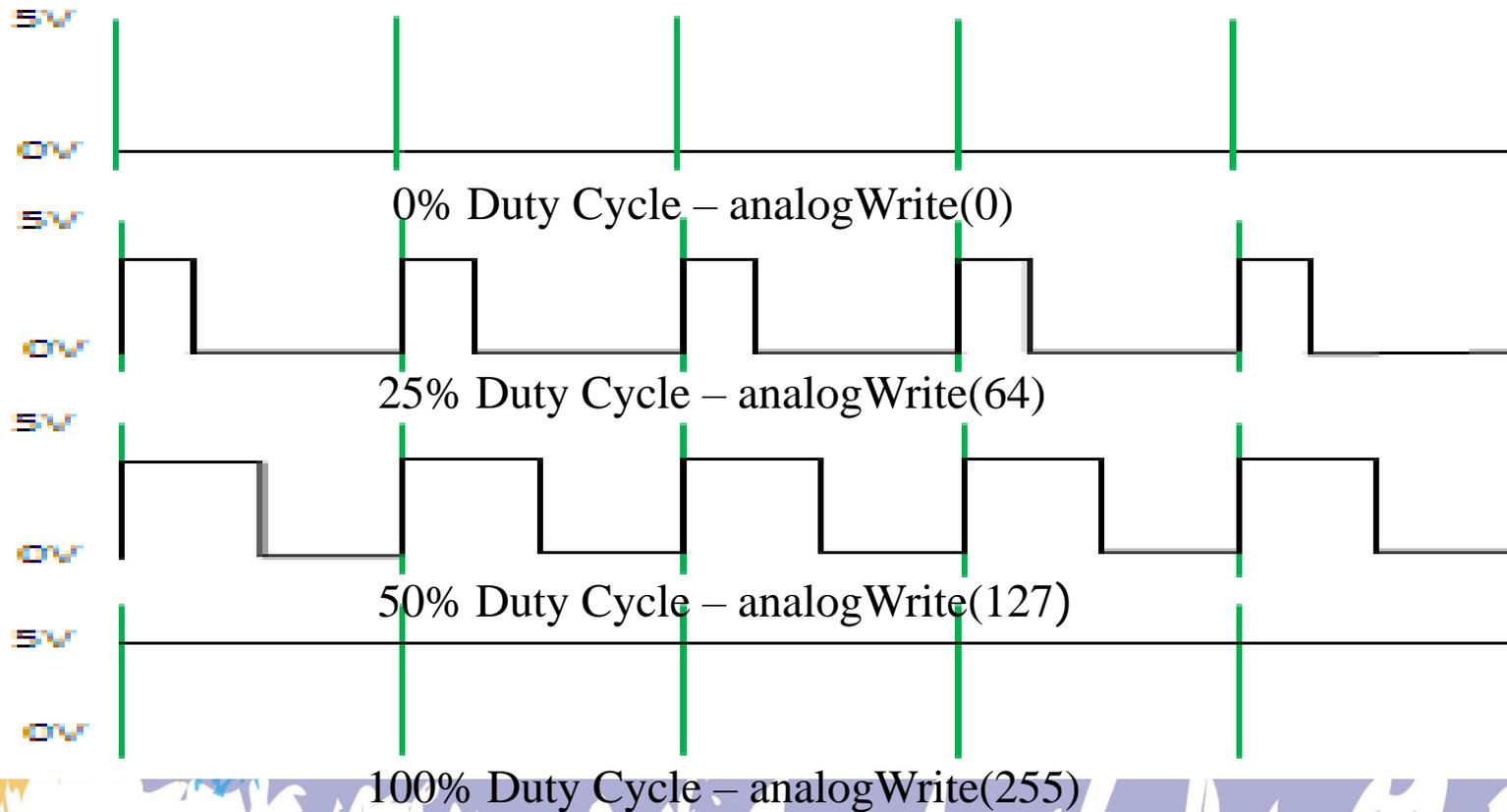
# 感光控制燈

- 本次實驗，以一個光敏電阻進行實驗。光敏電阻既可以根據光的強度改變電阻值的元件，即會需要模擬輸入讀取模擬值，本實驗可以藉由PWM接口實驗，將電壓換成光敏電阻當光線強度不同時，使得LED小燈的亮度也會有相對應的變化。

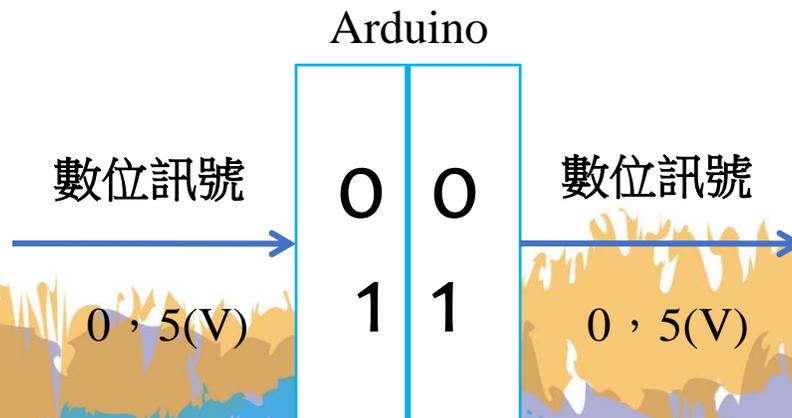
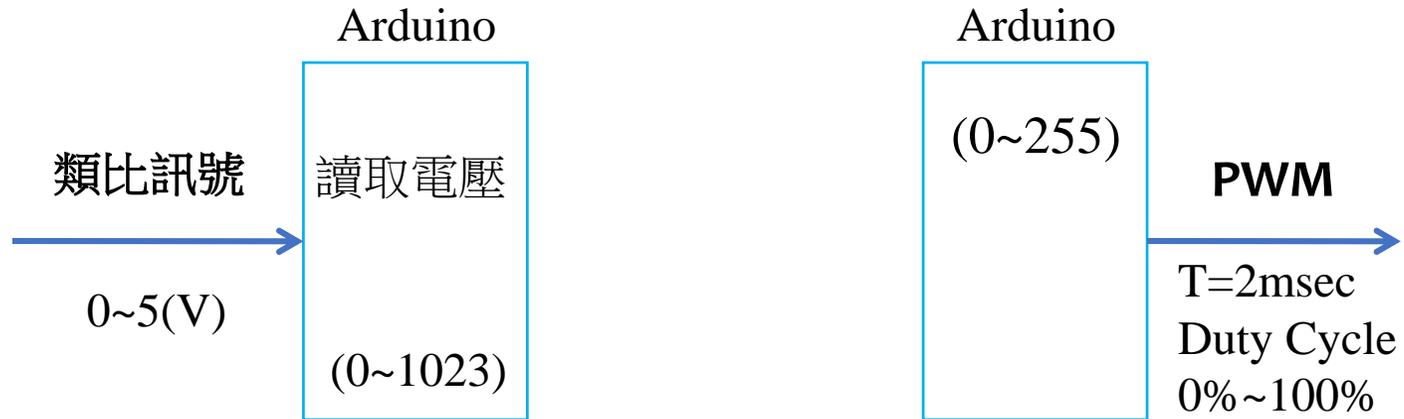


# 使用PWM產生類比訊號

1. 因我們可使用調整PWM(脈衝寬度調變)訊號產生輸出類比訊號，我們控制HIGH的時間來調整每個周期中的Duty Cycle。
2. (Arduino pin 上基本的切換頻率固定為 500Hz，每個週期是 2ms)，
3. 輸出的電壓是由ON 和OFF時間的平均值來得到，那我們就可以數位輸出類比訊號，**analogWrite**只接受0~255的數值

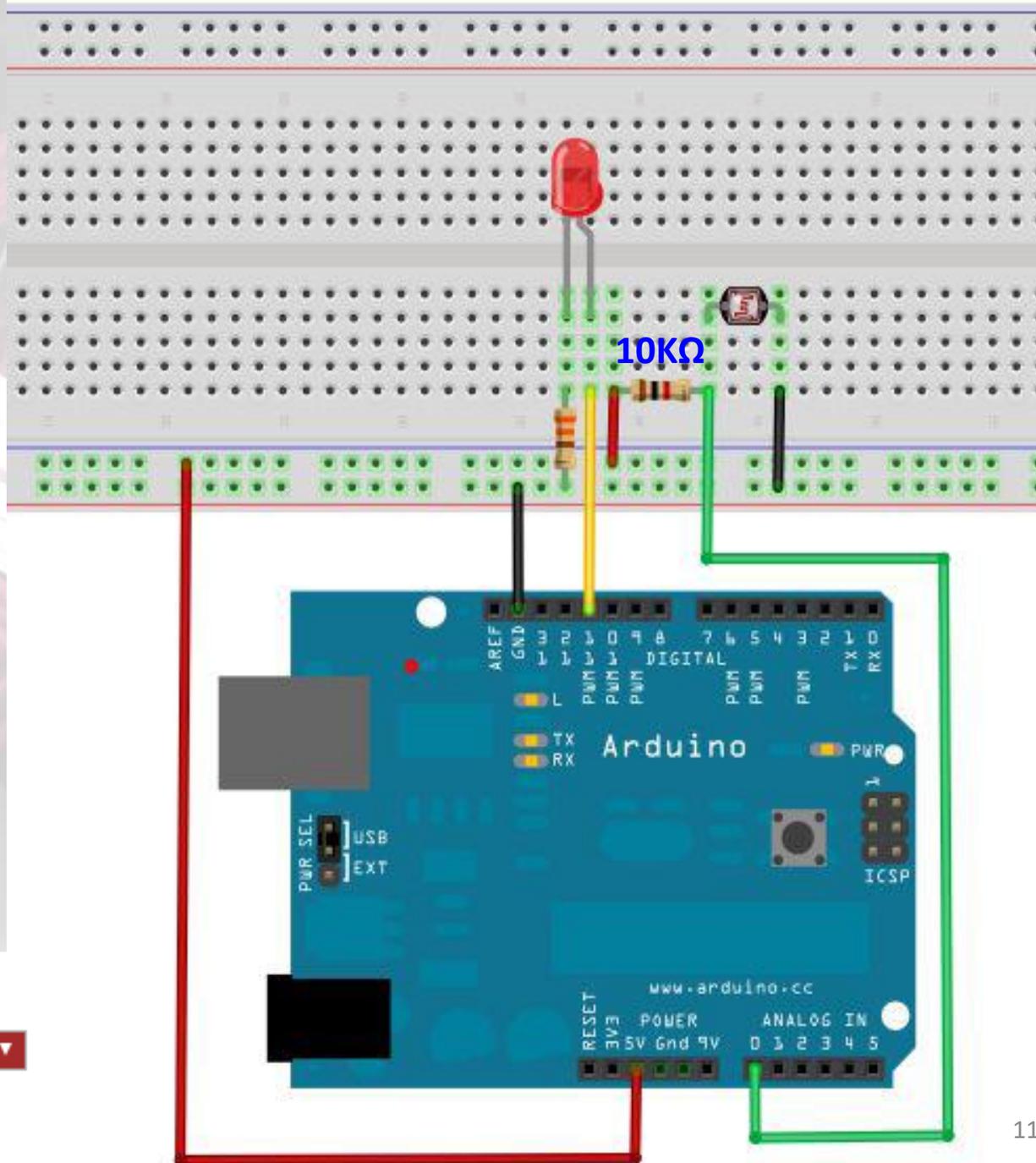


# 數位與類比的對應值



# 準備材料

- 光敏電阻 \* 1
- 紅色LED \* 1
- 10K $\Omega$ 電阻 \* 1
- 330 $\Omega$ 電阻 \* 1
- 麵包板 \* 1
- 麵包板跳線 \* 少許



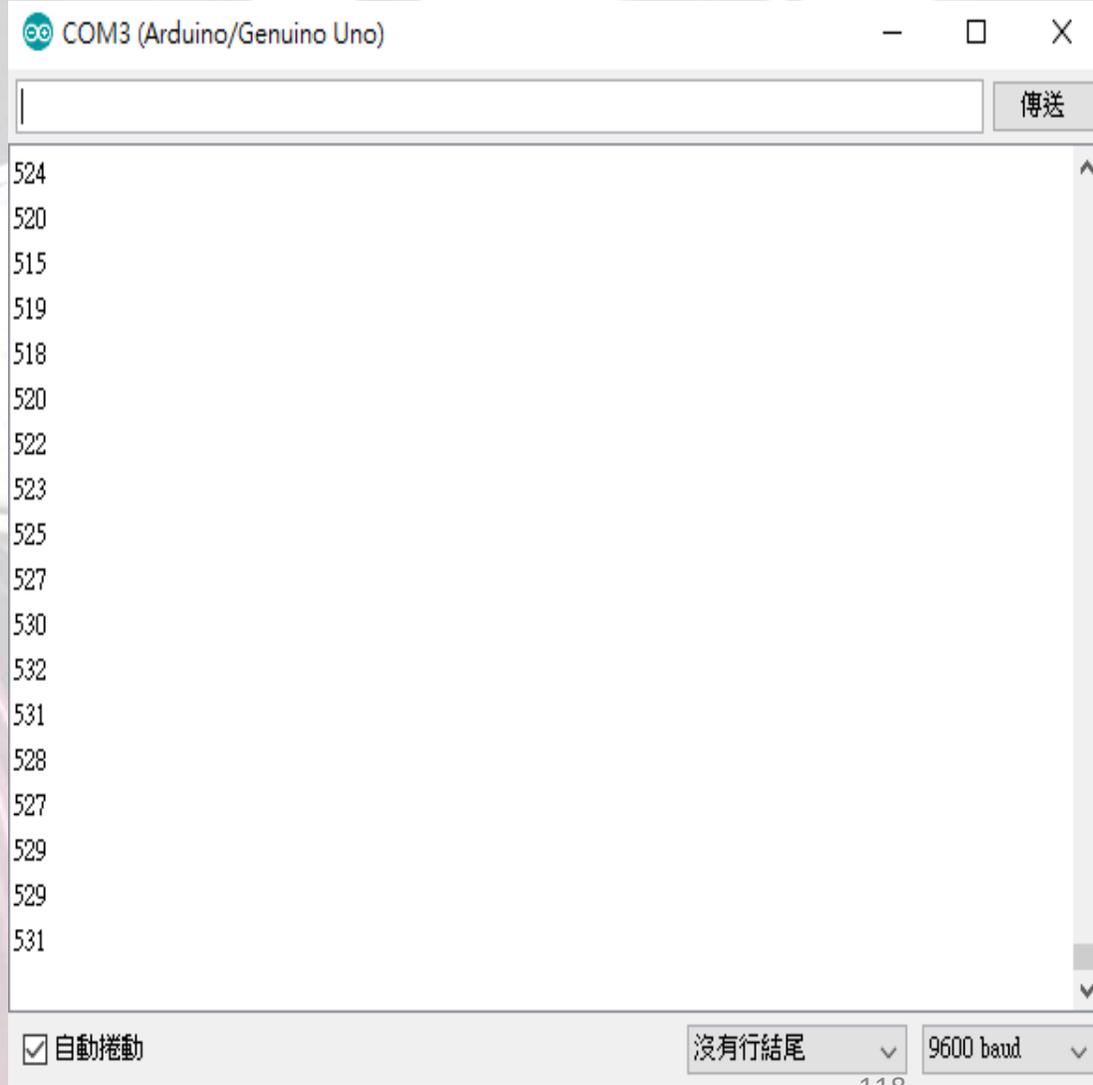
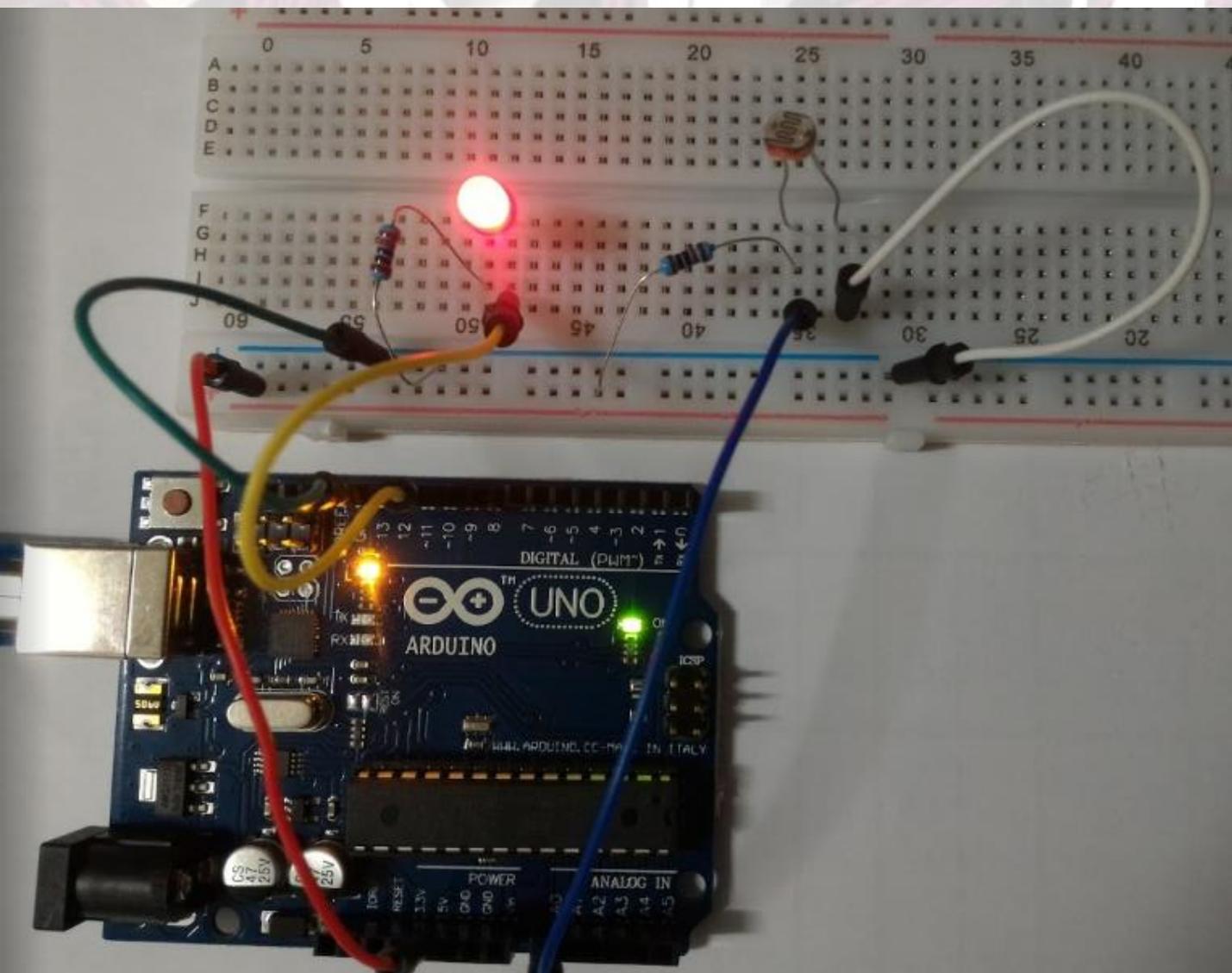
選擇電阻上各個色環的顏色：

棕色 ▼ 黑色 ▼ 黑色 ▼ 紅色 ▼ 棕色 ▼

電阻值:

- `int potpin=0;`//定義類比pin為A0 連接光敏電阻
- `int ledpin=11;`//定義數位(PWM)pin為11 藉由輸出PWM 來調節LED 亮度
- `int val=0;`//定義變數
- `void setup()`
- `{`
- `pinMode(ledpin,OUTPUT);`//定義數位(PWM)pin11為輸出
- `Serial.begin(9600);`//設置波特率為9600
- `}`
- `void loop()`
- `{`
- `val=analogRead(potpin);`//讀取光敏電阻的值給val
- `Serial.println(val);`//顯示即時觀測視窗
- `analogWrite(ledpin,val);`// 打開ledpin ( PWM 輸出最大值255 )
- `delay(100);`//延時0.1秒
- `}`

# 感光控制燈





light\_sensor

```
int potpin=0;//定義類比pin為A0 連接光敏電阻
int ledpin=11;//定義數位(PWM)pin為11 藉由輸出PWM 來調節LED 亮度
int val=0;//定義變數

void setup()
{
  pinMode(ledpin,OUTPUT);//定義數位(PWM)pin11為輸出
  Serial.begin(9600);//設置波特率為9600
}
void loop()
{
  val=(analogRead(potpin)-400)/2;//讀取光敏電阻的值給val
  Serial.println(val);//顯示即時觀測視窗
  analogWrite(ledpin,val);// 打開ledpin (PWM 輸出最大值255)
  delay(100);//延時0.1秒
}
```

上傳完畢。

草稿碼使用了 2398 bytes (7%) 的程式儲存空間。上限為 32256 bytes。  
全域變數使用了 188 bytes (9%) 的動態記憶體，剩餘 1860 bytes 給區域變數。上限為 2048 bytes。

傳送

14  
12  
12  
10  
10  
9  
9  
8  
8  
8  
8  
9  
9  
10  
11  
12  
14  
15  
15  
15  
15  
15  
15  
15  
14  
13  
12  
11  
10 自動捲動

沒有行結尾

9600 baud

119



## PWM-Light3

```
int potpin=0;//定義類比pin為A0 連接光敏電阻
int ledpin=11;//定義數位(PWM)pin為11 藉由輸出PWM 來調節LED 亮度
int val=0;//定義變數

void setup()
{
  pinMode(ledpin,OUTPUT);//定義數位(PWM)pin11為輸出
  Serial.begin(9600);//設置波特率為9600
}
void loop()
{
  val=analogRead(potpin);//讀取光敏電阻的值給val
  Serial.println(val);//顯示即時觀測視窗
  if (val >= 600)
  {
    digitalWrite(ledpin,HIGH);// 打開ledpin (PWM 輸出最大值255)
  }
  else
  {
    digitalWrite(ledpin,LOW);// 打開ledpin (PWM 輸出最大值255)
  }
  delay(100);//延時0.1秒
}
```

上傳完畢。

傳送

```
718
721
726
721
687
662
621
631
644
650
657
667
671
680
682
688
693
696
697
657
633
611
660
671
679
688
```

 自動捲動

沒有行結尾

120  
9600 baud



PWM-Light4

```
{
pinMode(ledpin,OUTPUT);//定義數位(PWM)pin11為輸出
Serial.begin(9600);//設置波特率為9600
}
void loop()
{
  val=analogRead(potpin);//讀取光敏電阻的值給val
  Serial.println(val);//顯示即時觀測視窗
  if (val >= 600)
  {
    //digitalWrite(ledpin,HIGH);// 打開ledpin (PWM 輸出最大值255)
    analogWrite(ledpin, 0); // LED 不亮
    delay (1000);
    analogWrite(ledpin, 31); // LED 亮12.5%
    delay (1000);
    analogWrite(ledpin, 63); // LED 亮25%
    delay (1000);
    analogWrite(ledpin, 95); // LED 亮37.5%
    delay (1000);
    analogWrite(ledpin, 127); // LED 亮50%
    delay (1000);
    analogWrite(ledpin, 159); // LED 亮62.5%
    delay (1000);
    analogWrite(ledpin, 191); // LED 亮75%
    delay (1000);
    analogWrite(ledpin, 223); // LED 亮87.5%
    delay (1000);
    analogWrite(ledpin, 255); // LED 亮100%
    delay (1000);
  }
  else
  {
    digitalWrite(ledpin,LOW);// 關閉ledpin (PWM 輸出最小值0)
  }
  delay(100);//延時0.1秒
}
```

傳送

```
374
375
375
377
377
380
381
377
370
366
367
373
378
380
383
384
388
389
390
390
391
392
388
378
373
361
366
368
371
372
373
375
375
375
377
376
377
373
369
```

● 中午休息下課囉!!!