



Communication & Biologic
Integrated Circuit
Guard Patch Alliance

生醫穿戴式裝置開發課程 模組

健康貼身守護神小聯盟



簡報人：李順裕 教授

健康貼身守護神小聯盟召集人(共同主持人)

教育部智慧空間電子應用聯盟中心:智慧感測
系統與雲端整合運用課程開發與推廣召集人

國立成功大學電機系

通訊暨生物積體電路設計實驗室

李順裕 教授

實驗室: Communication and Biologic ICs (CBIC)
(通訊暨生物積體電路設計實驗室)

產品規劃: 貼身守護神

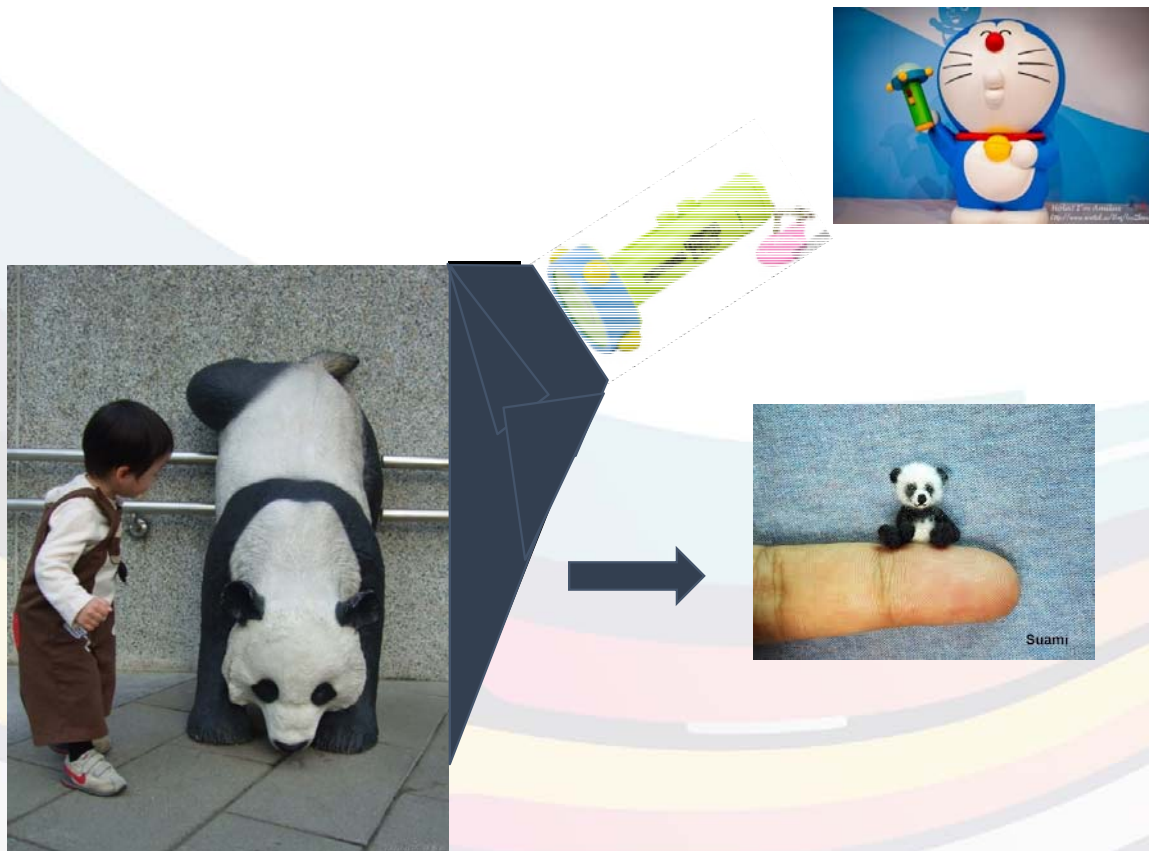
- 體外: 智慧穿戴系統晶片
- 體內: 神經調控系統晶片

夢想: 醫電園
(半導體產業+醫療儀器產業+通訊網路產業)

哆啦A夢

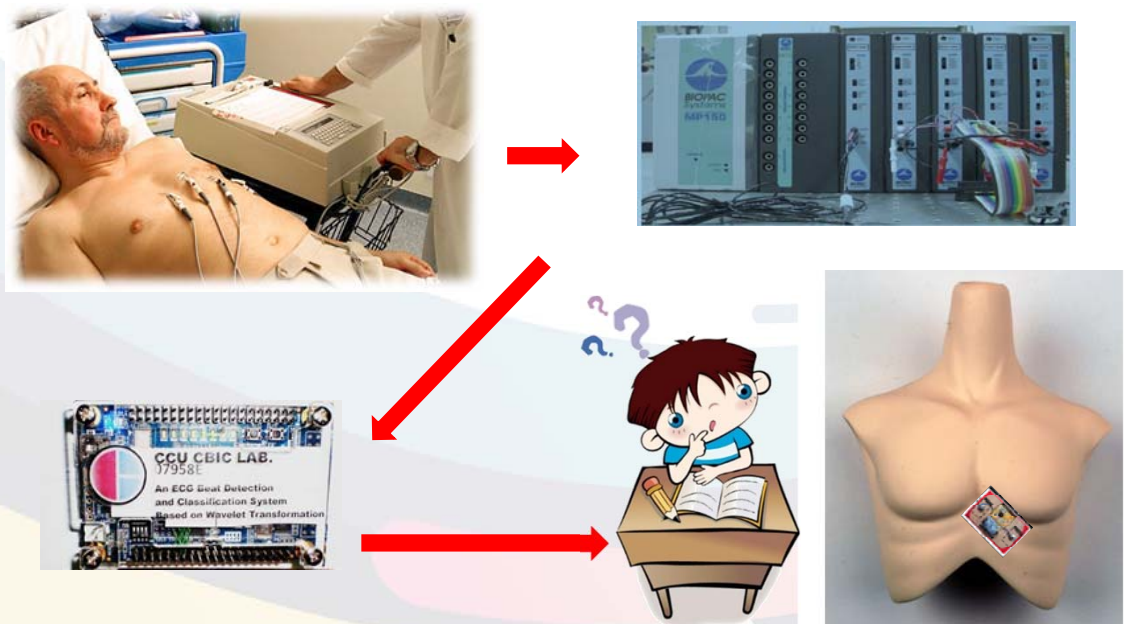


哆啦A夢



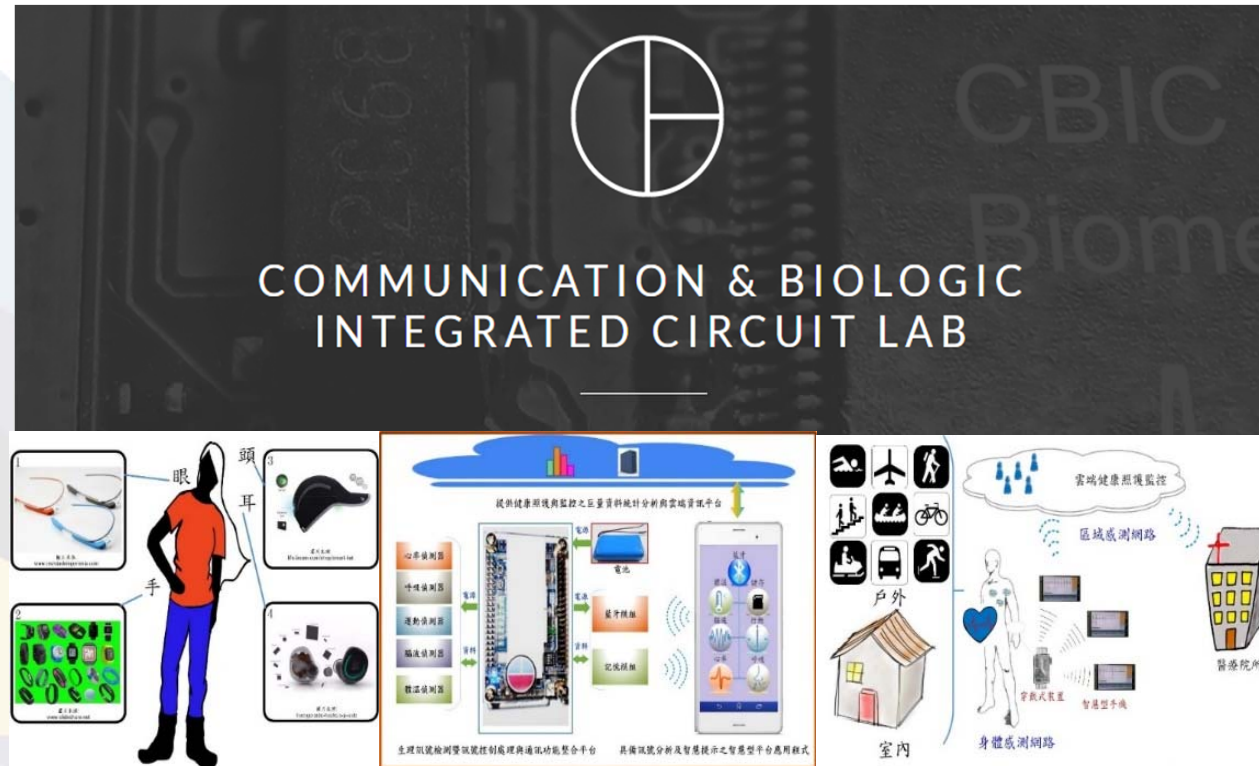
產品情境

- 改變醫療環境囉!



- 案例: 林昭維醫師: 我的病人在阿里山爬山, 覺得胸口悶痛, 是否可將他的ECG(至少三導程)即時傳到我的手機或辦公室

通訊暨生物積體電路設計(CBIC)實驗室



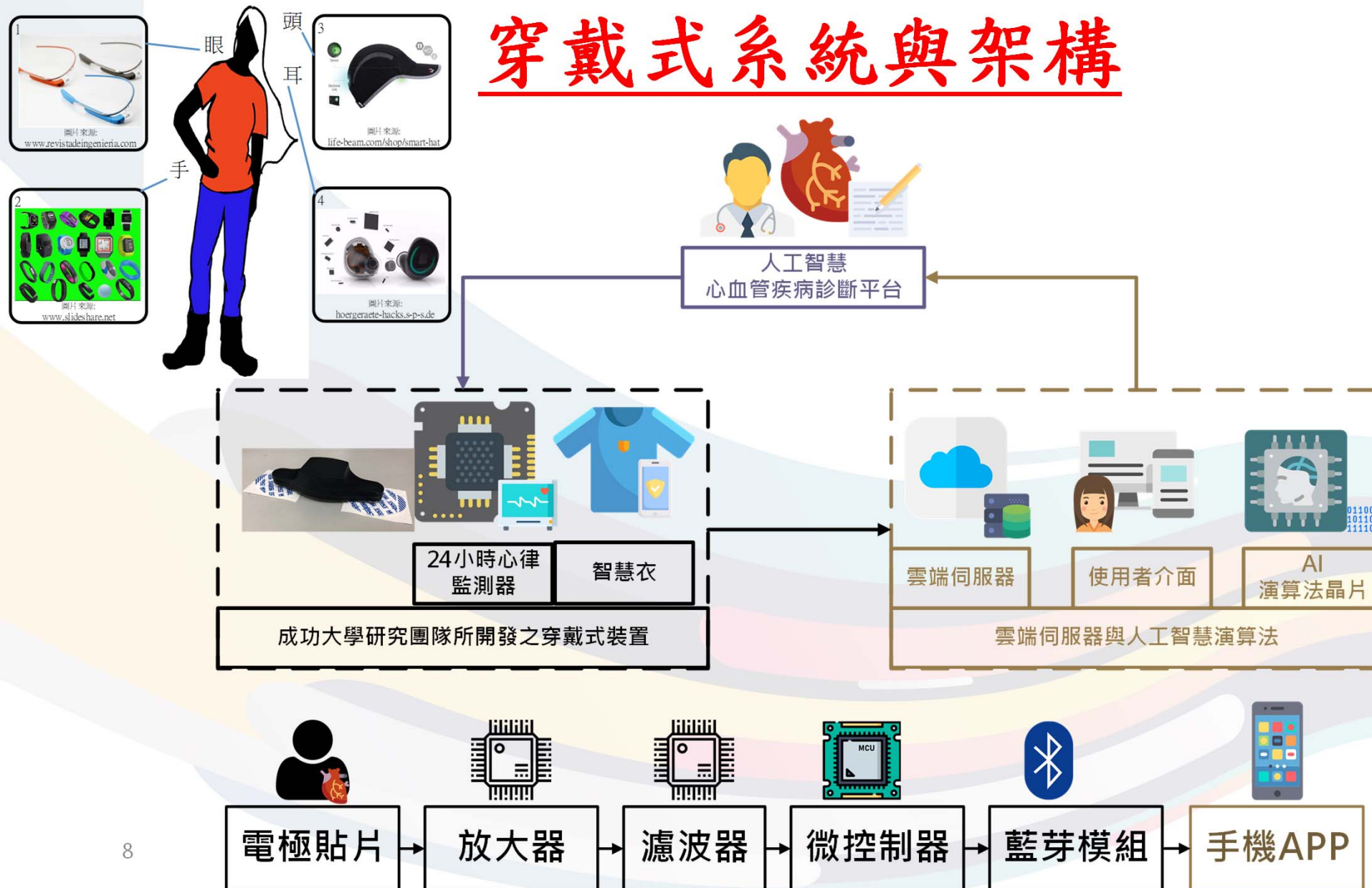
參考網址

<https://www.youtube.com/YutechCBIC>

生理訊號監測系統與平台



穿戴式系統與架構



科技技術:貼身守護神

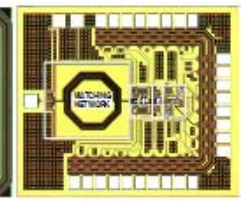
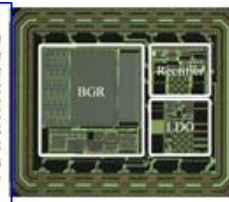
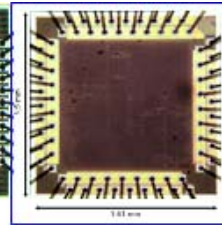
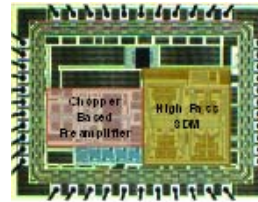
- 低功耗與高解析度晶片:長時穿戴
- 穿戴模組製作與驗證:安全便利
 - 通過IEC電性安規認證
 - 長 7.4公 X 寬 3.4公分 X 高 1.4公分
- 手機軟體服務與測試:健康保障
 - 與成大醫院共同研發心律診斷APP
 - 通過成大醫院人體試驗審查(IRB)
 - 已進行超過50位病人測試

類比前端

訊號處理

電源管理

無線傳輸



產品介紹影片



產品應用(10產品)

- 貼身守護神:24小時無線心律監控晶片系統(技術轉移:新創公司美新華股份有限公司、人工智慧X物聯網創新創業競賽第一名)
- 智慧衣(合作夥伴:天疆實業)+體內心電還原技術(合作夥伴:工研院材化所)
- 寵物衣(合作夥伴:華育機電有限公司、高雄寵物醫院、大學部國際競賽第一名)
- 阻抗偵測技術(自行開發)(體脂肪偵測)(南良集團)
- 智慧醫療聽診器(南部智慧生醫產業聚落推動計畫)
- 癲癇腦波偵測與刺激系統(SPARK TAIWAN (台灣生醫與醫材轉譯加值人才培訓計畫))
- 智慧穿戴式物聯網開發模組(科技部計畫科國司、教育部智慧聯網技術與應用人才培育計畫)
- 車用電子疲勞偵測系統(大學部國際競賽亞洲代表隊)
- 智慧嬰兒包巾(科技部運用法人鏈結產學合作計畫績優團隊第二名)
- 具可攜式無線尿液檢測與預防心血管疾病之晶片系統與平台(「智慧終端半導體製程與晶片系統研發專案計畫(半導體射月計畫)」)



科技技術介紹

智慧無線穿戴式系統 四大技術介紹

- I. 物聯網傳輸&接收技術
- II. 穿戴式電路設計
- III. 無線生醫晶片系統設計
- IV. 應用軟體開發

Guard Patch: Wearable Devices

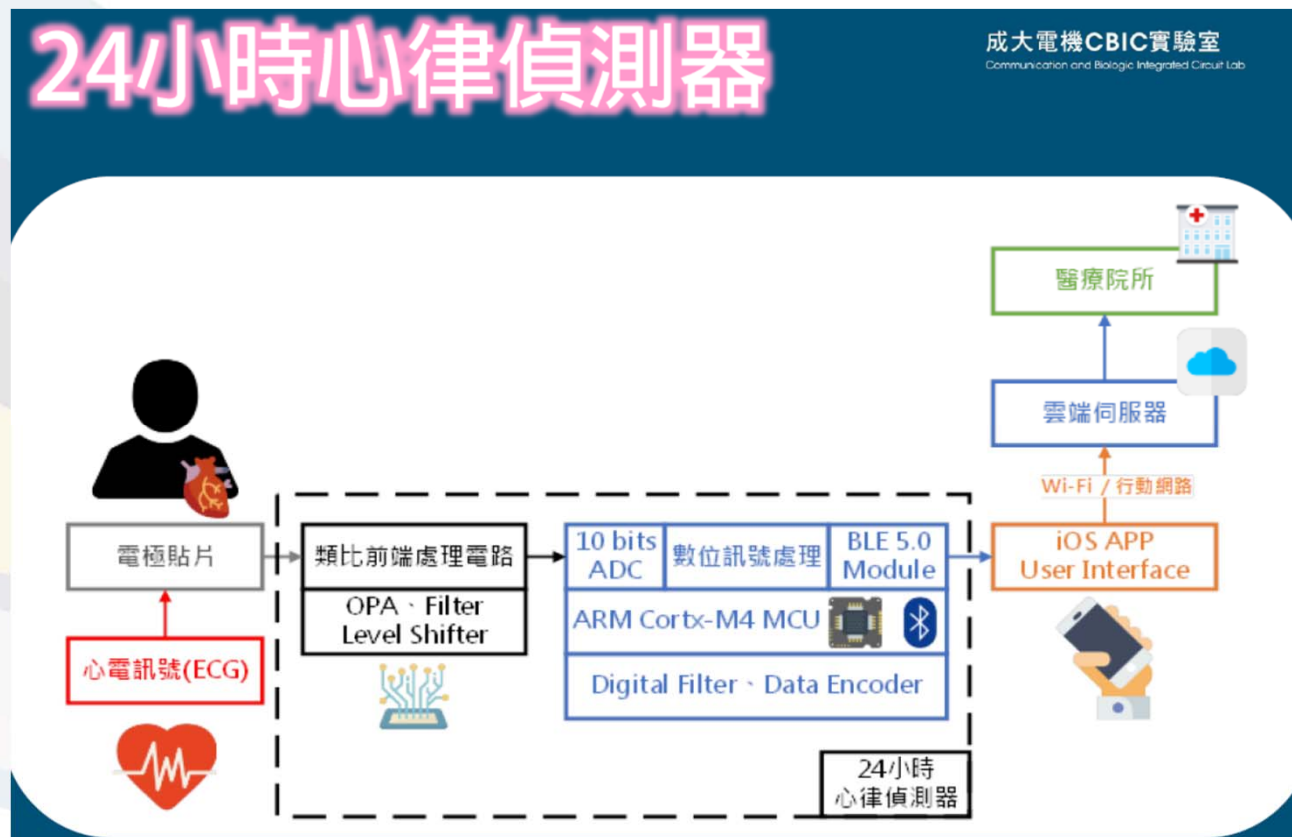
貼身守護神:穿戴式產品介紹



Guard Patch: Wearable Devices

貼身守護神:穿戴式產品介紹 (產品-1)

- 24-hour Electrocardiogram (ECG) Patch (貼身守護神:24小時無線心律監控晶片系統)－貼心片



Guard Patch: Wearable Devices

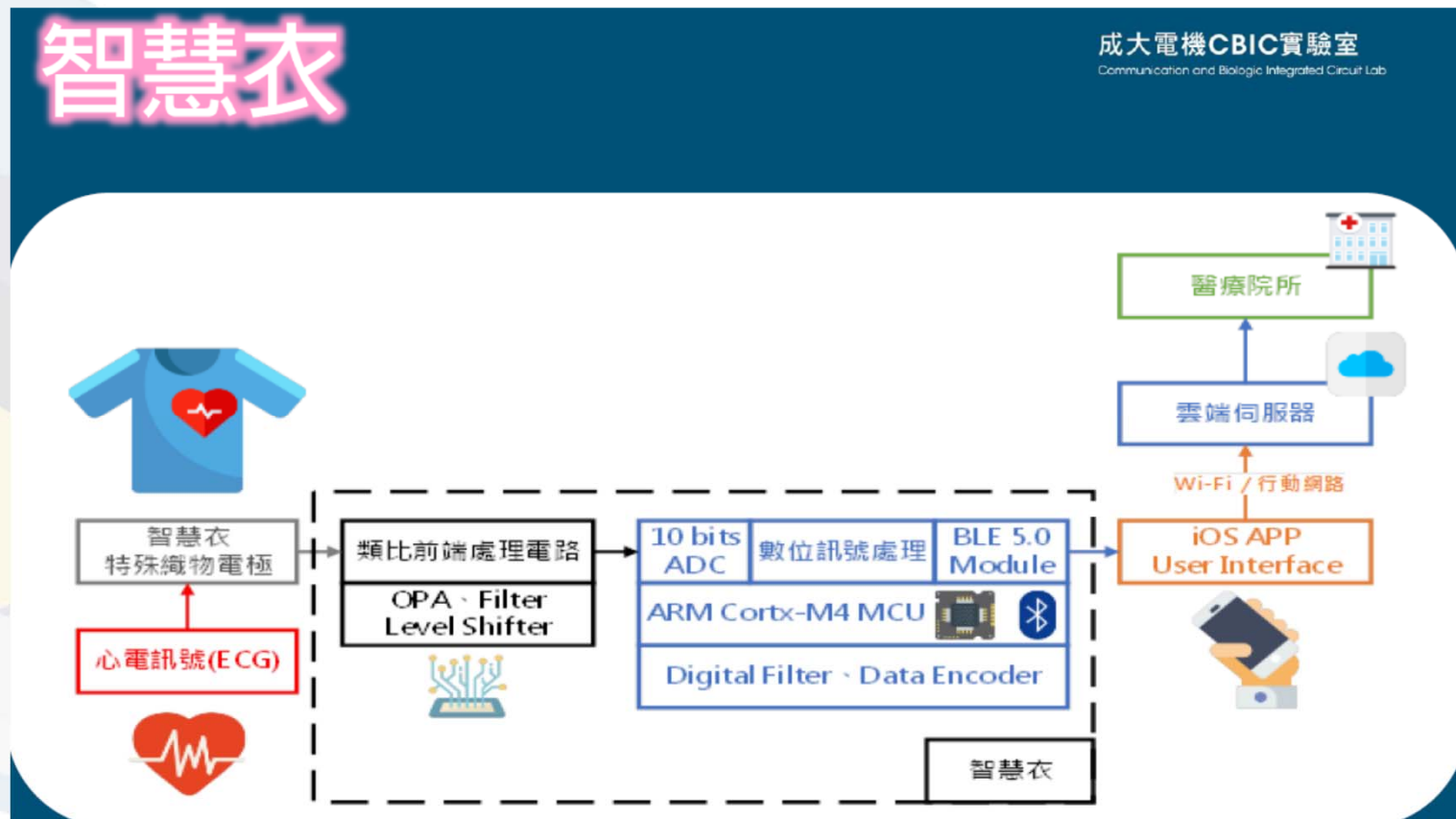
貼身守護神:穿戴式產品介紹 (產品-1)



Guard Patch: Wearable Devices

貼身守護神:穿戴式產品介紹 (產品-2)

■ Smart Clothes (智慧衣) – 貼心衣



智慧衣技術介紹:衣服

聚陽實業



BeatInfo(運動用)

以心跳為主



智慧衣技術介紹:衣服

三司達 (2PIR x Caballero)



智能感測運動背心

以心跳為主

智慧衣技術介紹:衣服

萬九科技



Cardiosport 電路感測器:

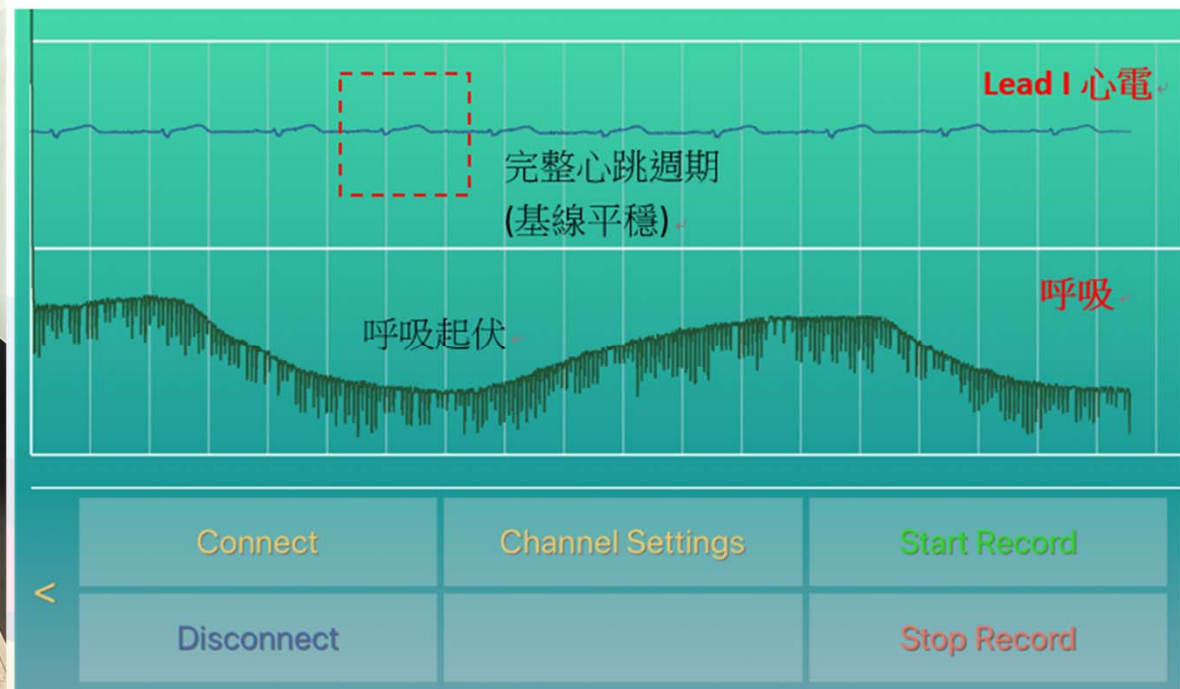
- 1 - 3 Lead ECG Recorder
- 2 Channel differential ECG inputs
- Event Detection
- HRV
- Respiratory Rate
- Vital Signs
- 7-14 days battery life
- ASIC



主要提供感測電路模組

智慧衣生理訊號檢測技術:衣服

成功大學CBIC實驗室 (天疆實業:經濟部CITD & SBIR計畫)



智慧衣技術介紹:人因工程分析

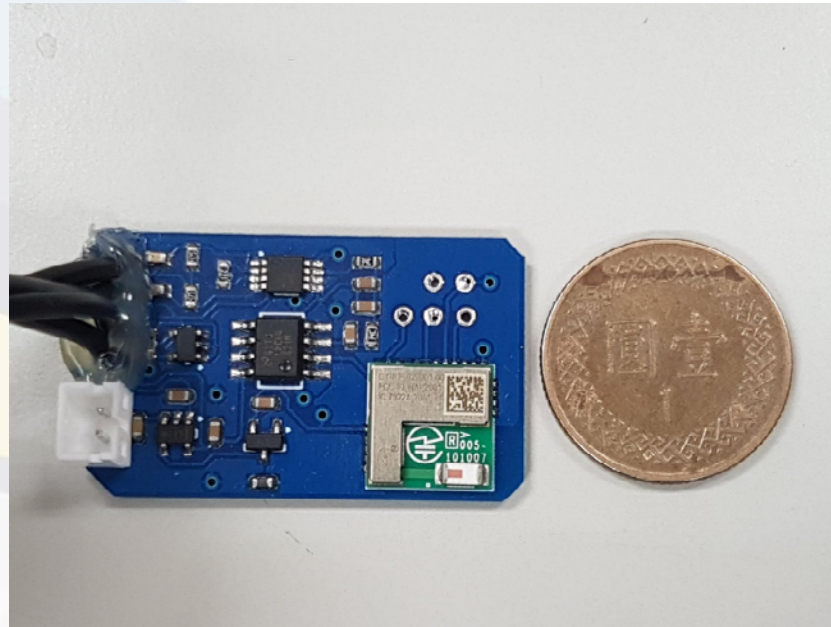
成功大學CBIC實驗室（移動測試、水洗測試、舒適性測試）



智慧衣生理訊號檢測技術:檢測技術

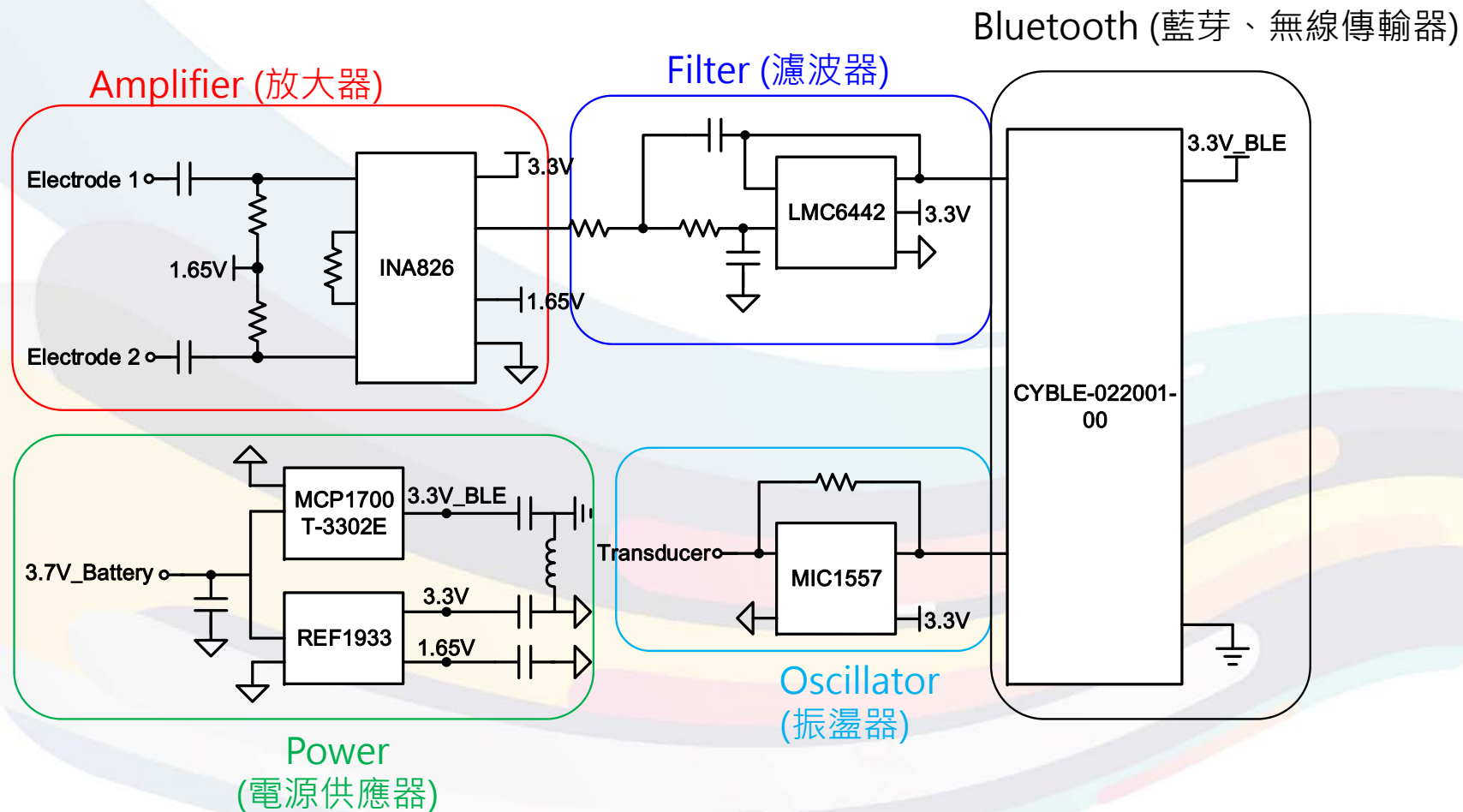
■ System overview

- Size = 2.1cm(W)* 3.5cm(L)*8mm(H)
- Weight depends on Li-ion battery
- Power = 3.7V ~ 4.2V(Li-ion battery)



智慧衣生理訊號檢測技術:檢測技術

■ System block



Guard Patch: Wearable Devices

貼身守護神:穿戴式產品介紹 (產品-2)

■ 硬體製作

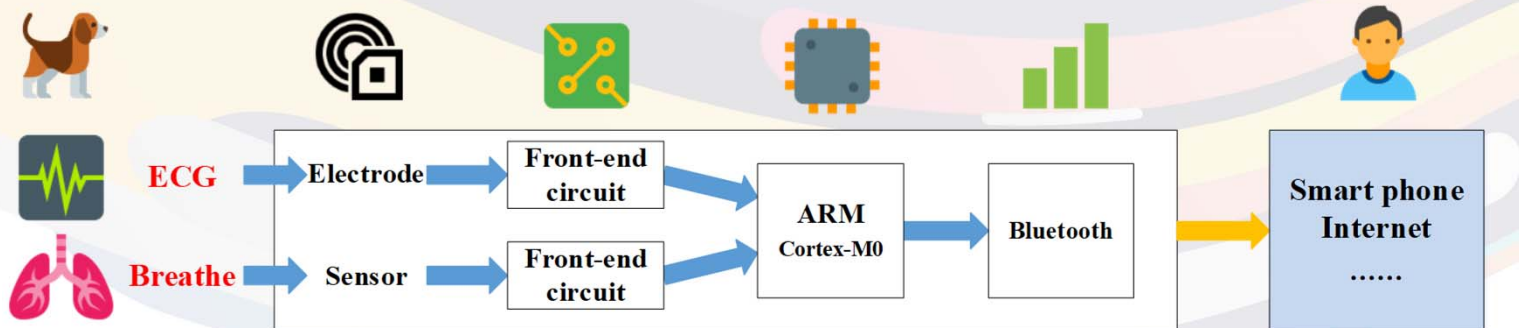
■ 軟體服務:使用者介面

網路服務



■ 參考產品展示網頁:youtube.com/Innocbic

Smart clothing of pet



帛帥企業有限公司
<http://www.smpet.com.tw>

寵物衣影片介紹



現在看到狗狗寶正在偵測狗狗的生理訊號

寵物衣世界競賽冠軍影片介紹

Smart Pet Clothing

Guard Patch: Wearable Devices

貼身守護神:穿戴式產品介紹 (產品-4)

■ 阻抗偵測技術 (體脂肪偵測、癌細胞偵測)

- 運動員體適能評估
- 減重、體脂計(BMI)、營養學
- 人體體液狀態監測(水分、血液)
 - 急性心衰竭患者
 - 慢性腎臟病患者
- 心肺功能評估
- 呼吸參數量測

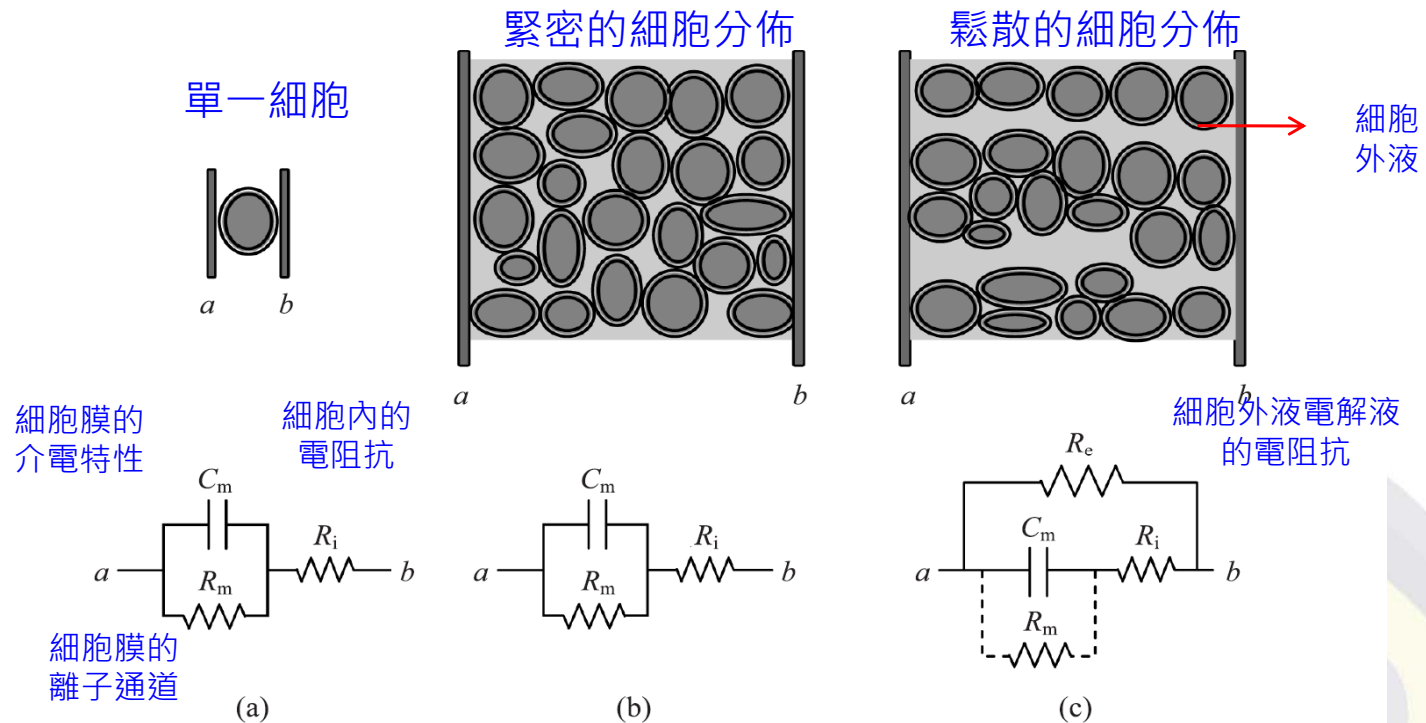


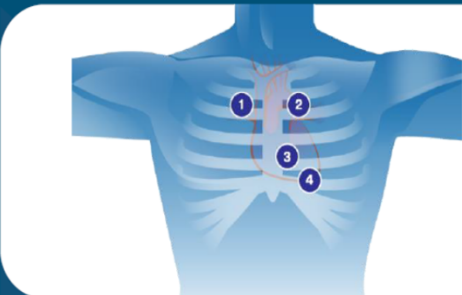
圖 2.33 生理組織電阻抗的簡單電路模型。(a) 電流流經單一細胞時的電路模型；(b) 緊密的細胞分佈的情形下之電路模型；(c) 電流路徑包括細胞外的電解液時的電路模型

Guard Patch: Wearable Devices

貼身守護神:穿戴式產品介紹 (產品-5)

Smart Stethoscope (智慧醫療聽診器) – 貼心音

智慧聽診器 成大電機CBIC實驗室
Communication and Biologic Integrated Circuit Lab



- 1 Base Right (Aortic area)**
Base right (aortic) is the second intercostal space to the right of the sternum. You can best hear sounds from the aortic valve in this area.
- 2 Base Left (Pulmonic area)**
Base left (pulmonic) is the second intercostal space to the left of the sternum. You can best hear sounds from the pulmonic valve in this area.
- 3 Lower Left Sternal Border (LLSB) (Tricuspid area)**
Left lower sternal border (tricuspid) is the fourth intercostal space to the left of the sternum. You can best hear tricuspid valve and right heart sounds in this area.
- 4 Apex (Mitral area)**
Apex (mitral) is the fifth intercostal space in the midclavicular line. It's easiest to hear mitral valve and left heart sounds in this area.

Four important areas used for cardiac auscultation



Guard Patch: Wearable Devices

貼身守護神:穿戴式產品介紹 (產品-5)

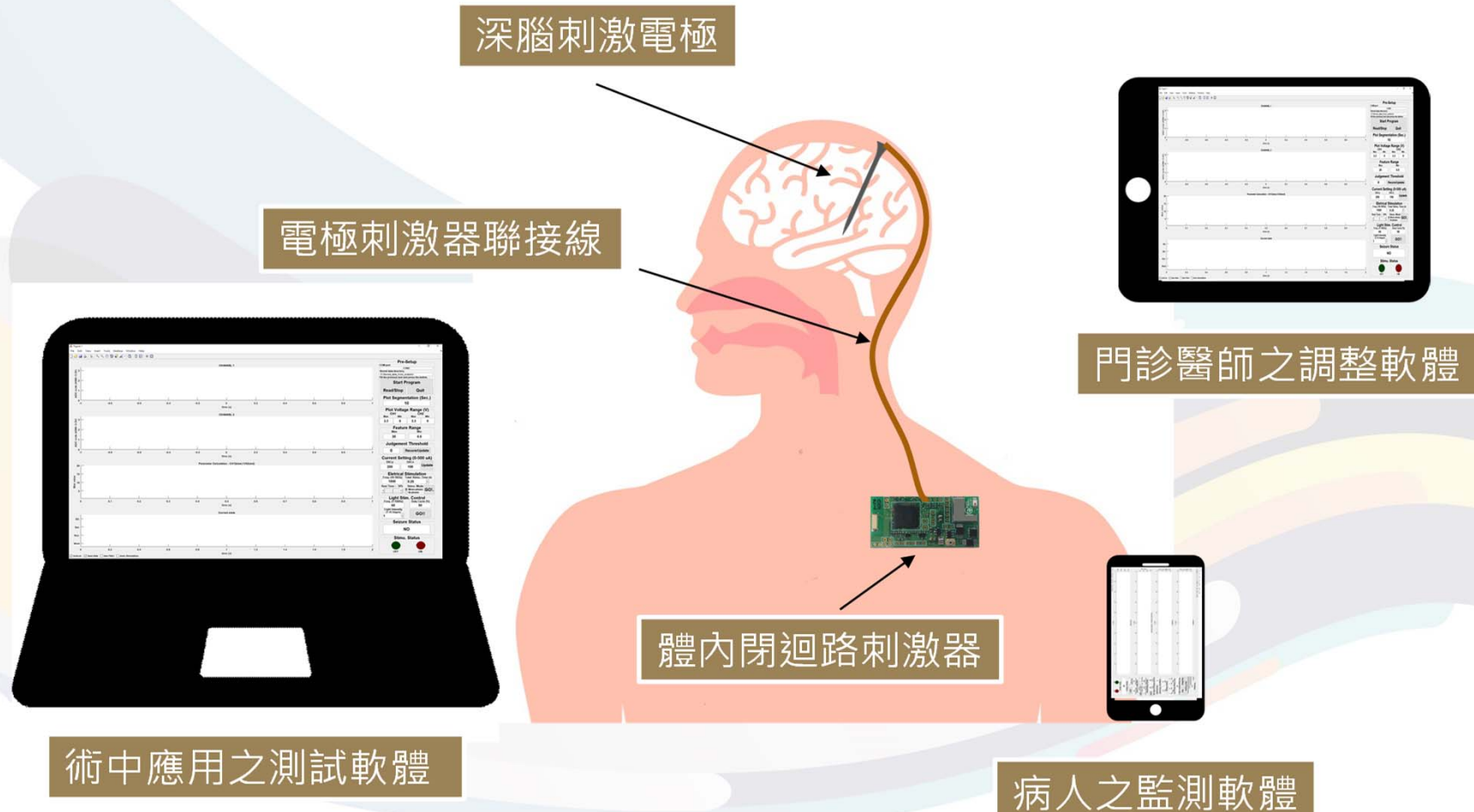
■ 世界觀摩展示



Guard Patch: Wearable Devices

貼身守護神:穿戴式產品介紹 (產品-6)

■ Epilepsy Detection and Stimulation (癲癇腦波偵測與刺激系統)



癲癇腦波偵測與刺激系統影片介紹



Guard Patch: Wearable Devices

貼身守護神:穿戴式產品介紹 (產品-7)

■ Trianswer (智慧穿戴式物聯網開發模組)



裕晶 醫學科技
股份有限公司

TriAnswer

Try and get your answer

TriAnswer(試穿戴)是一款生醫穿戴式裝置開發平台
實現您腦海中，每一個特別的創意

連結想法與真實的橋梁

- 直徑2.5公分，完成即為鞋型品
- 可處理十餘種生理訊號
- 高擷取訊號品質，利於後續分析
- 完整、開源的演算法函式庫
- 使用者可自行組裝



創意總在嘗試中發生

- 學生
- 創客
- 生醫應用開發商
- 任何想要動手DIY的人們



實現，並不困難

- 翻轉 你的學習力
- 光速 你的行動力
- 激發 你的想像力



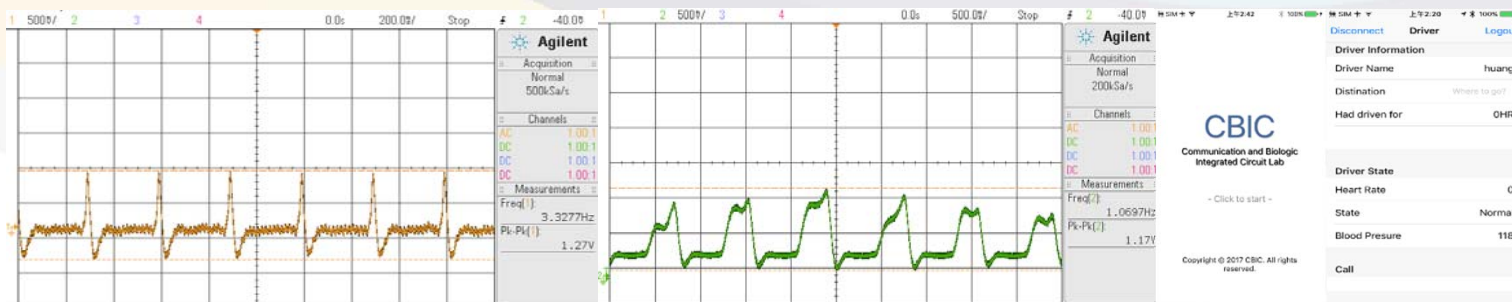
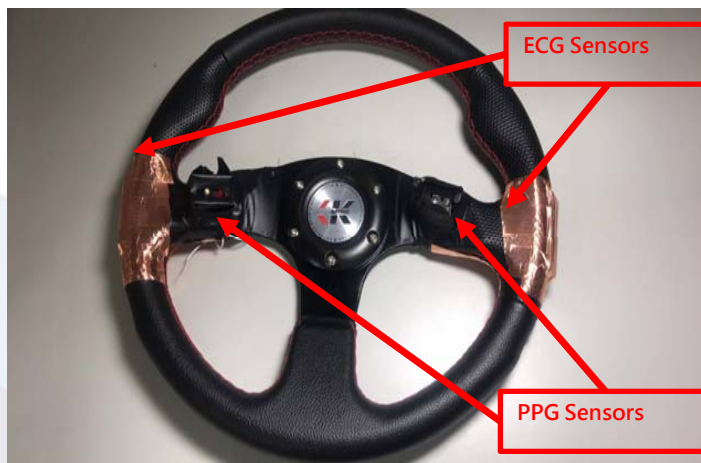
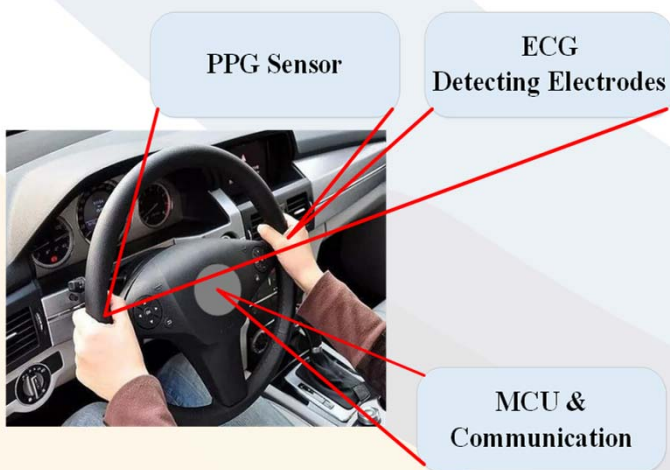
TriAnswer: Try and Get Your Answer (試穿戴)



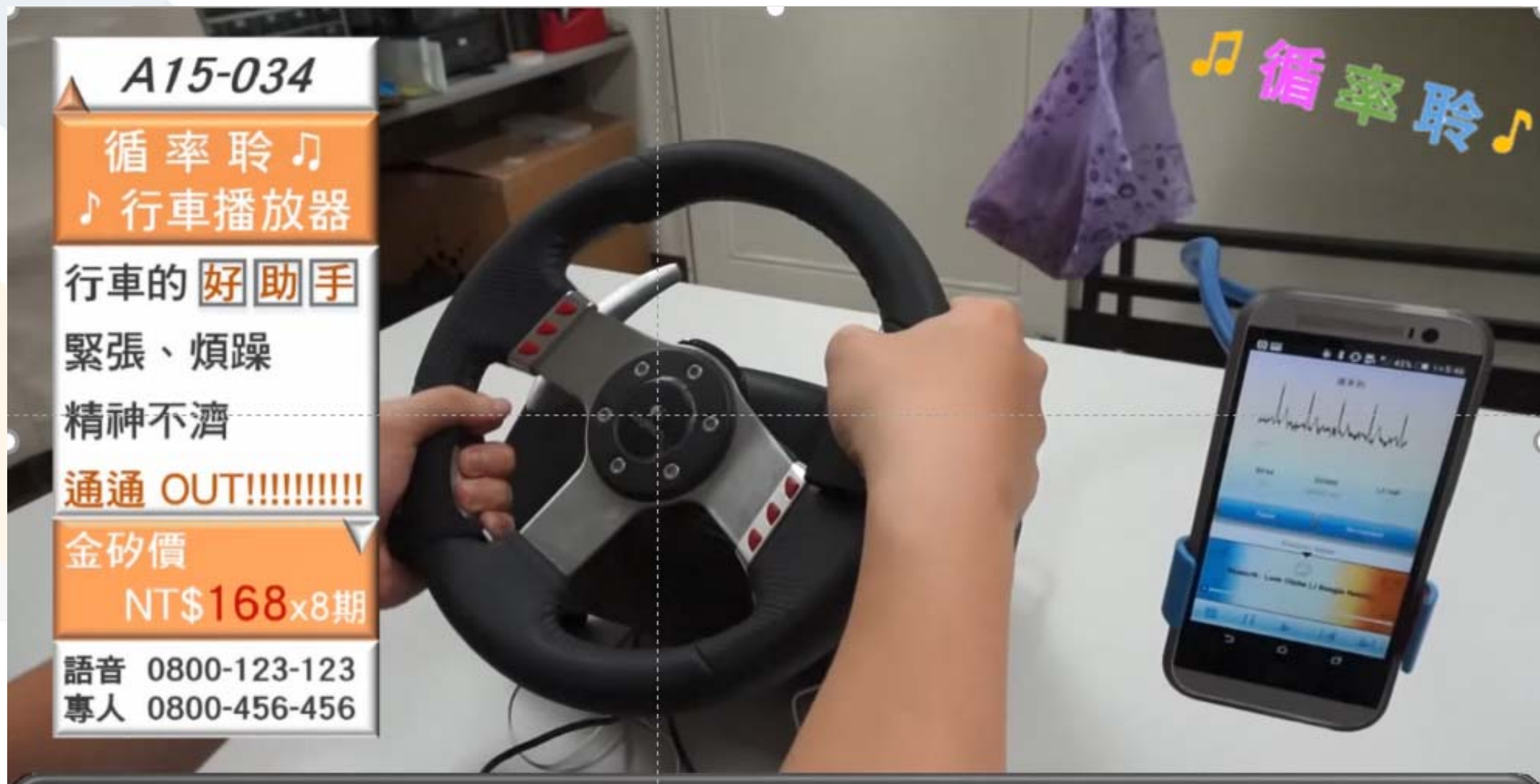
Guard Patch: Wearable Devices

貼身守護神:穿戴式產品介紹 (產品-8)

Car Driver's Fatigue Monitoring System (車用電子疲勞偵測系統)



旺宏金砂獎影片



A15-034

循率聆 ♪
♪ 行車播放器

行車的 **好** **助** **手**

緊張、煩躁
精神不濟
通通 OUT!!!!!!!!!!!!

金砂價
NT\$168x8期

語音 0800-123-123
專人 0800-456-456

♪ 循率聆 ♪

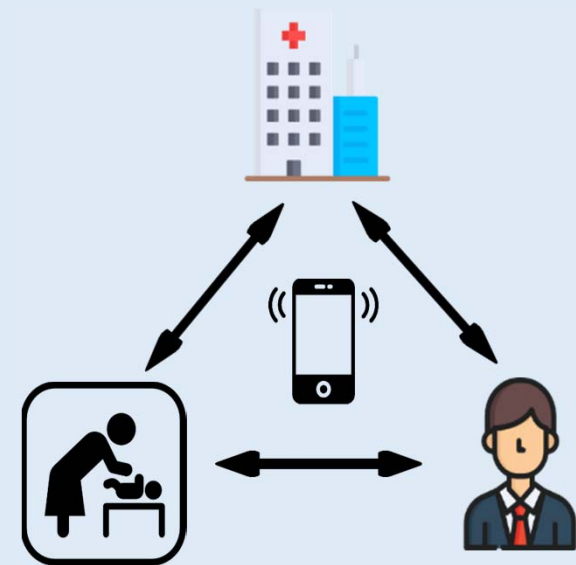
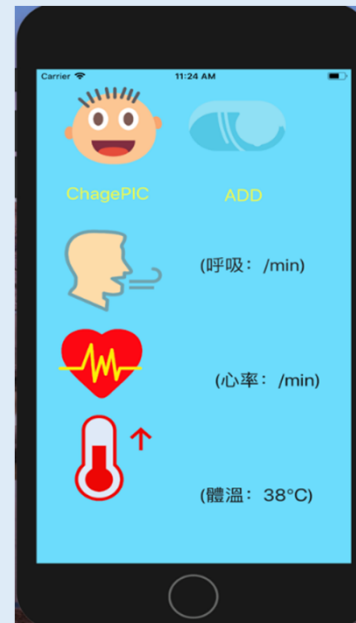
世界競賽展示影片



Guard Patch: Wearable Devices

貼身守護神:穿戴式產品介紹 (產品-9)

智慧嬰兒包巾



教育部智慧聯網技術與應用人才培育計畫

展示影片



Guard Patch: Wearable Devices

貼身守護神:穿戴式產品介紹 (產品- 10)

■ Portable and Wireless Urine Detection System and Platform (具可攜式無線尿液檢測與預防心血管疾病之晶片系統與平台)



2018未來科技展展示影片



結論:10項高科技產品應用

- 貼身守護神:24小時無線心律監控晶片系統(技術轉移:新創公司美新華股份有限公司、人工智慧X物聯網創新創業競賽第一名)
- 智慧衣(合作夥伴:天疆實業)+體內心電還原技術(合作夥伴:工研院材化所)
- 寵物衣(合作夥伴:華育機電有限公司、高雄寵物醫院、大學部國際競賽第一名)
- 阻抗偵測技術(自行開發)(體脂肪偵測)(南良集團)
- 智慧醫療聽診器(南部智慧生醫產業聚落推動計畫)
- 癲癇腦波偵測與刺激系統(SPARK TAIWAN (台灣生醫與醫材轉譯加值人才培訓計畫))
- 智慧穿戴式物聯網開發模組(科技部計畫科國司、教育部智慧聯網技術與應用人才培育計畫)
- 車用電子疲勞偵測系統(大學部國際競賽亞洲代表隊)
- 智慧嬰兒包巾(科技部運用法人鏈結產學合作計畫績優團隊第二名)
- 具可攜式無線尿液檢測與預防心血管疾病之晶片系統與平台(「智慧終端半導體製程與晶片系統研發專案計畫(半導體射月計畫)」)

人工智慧服務影片



心臟疾病衛教影片



2019_Afib衛教影片_國語_加片尾

哈囉! 我是小小醫學特派解說員!

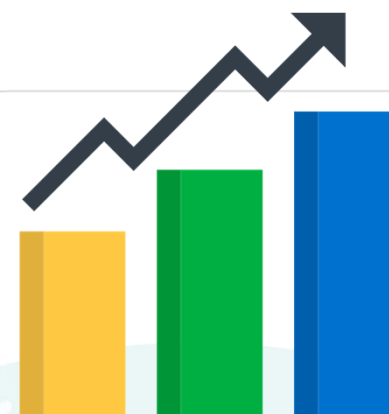
感謝各位聆聽

夢想很大(醫電園)
路還很長(守護神)



More Information

youtube.com/YutechCBIC



生醫穿戴式裝置開發課程模組

課程：Trianswer 模組介紹

Presenter : Tony助教

Date : 2020/01/30~2021/12/31





Outline

- 前言
 - 物聯網？生醫穿戴式裝置？
 - Why Trianswer？
- 生理訊號原理、Trianswer模組介紹
 - ECG(心電訊號)、EMG (肌電訊號)、PPG (光體積描述訊號)、太陽能充電、MCU (微控制器)
- Trianswer操作流程介紹



Outline

- 前言
 - 物聯網？生醫穿戴式裝置？
 - Why Trianswer？
- 生理訊號原理、Trianswer模組介紹
 - ECG(心電訊號)、EMG(肌電訊號)、PPG(光體積描述訊號)、太陽能充電、MCU(微控制器)
- Trianswer操作流程介紹



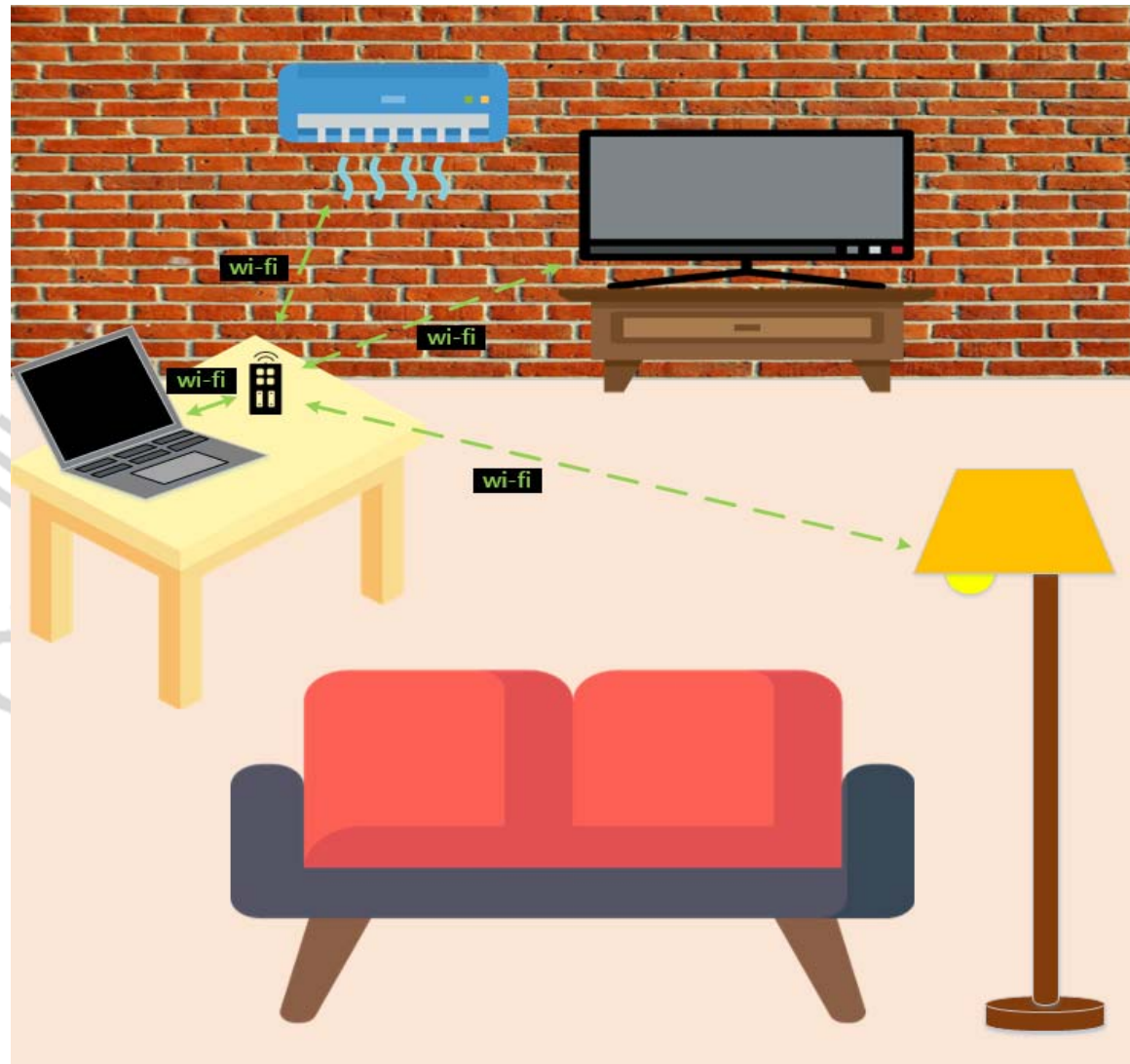
前言：物聯網

- 物聯網，Internet of Things
- 比爾·蓋茲在1995年出版的《未來之路》一書中提及物互聯
- 透過無線傳輸技術，將各個獨立的物品 (Things) 連結在一起
 - 形成一互相溝通的網絡
- 智慧家居、智慧農業環控、車聯網...等



智慧家居

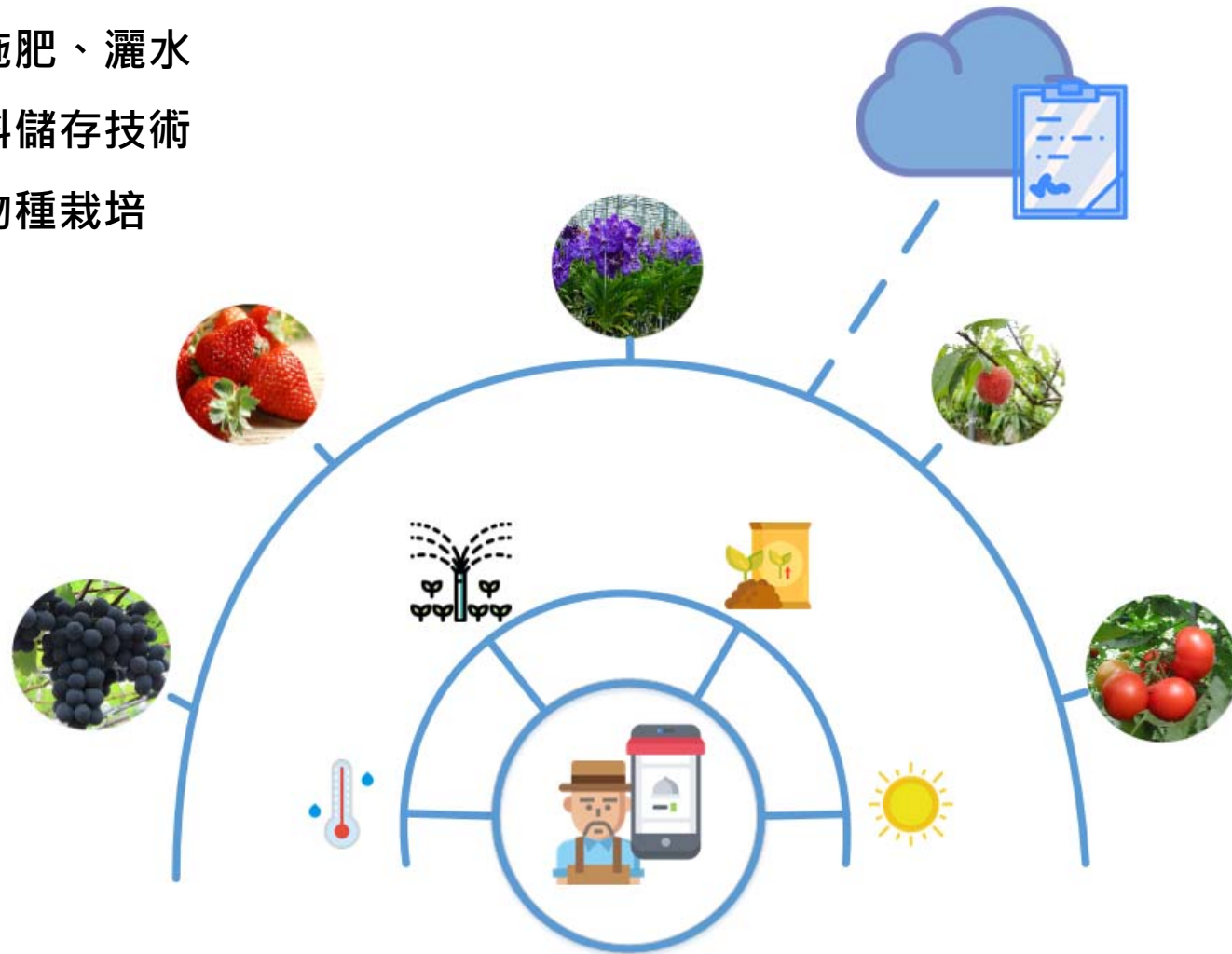
- 自動調節室內溫度
- 自動調節室內亮暗
- 自動關閉無使用之電器
- 偵測使用者習慣，選取
偏好節目





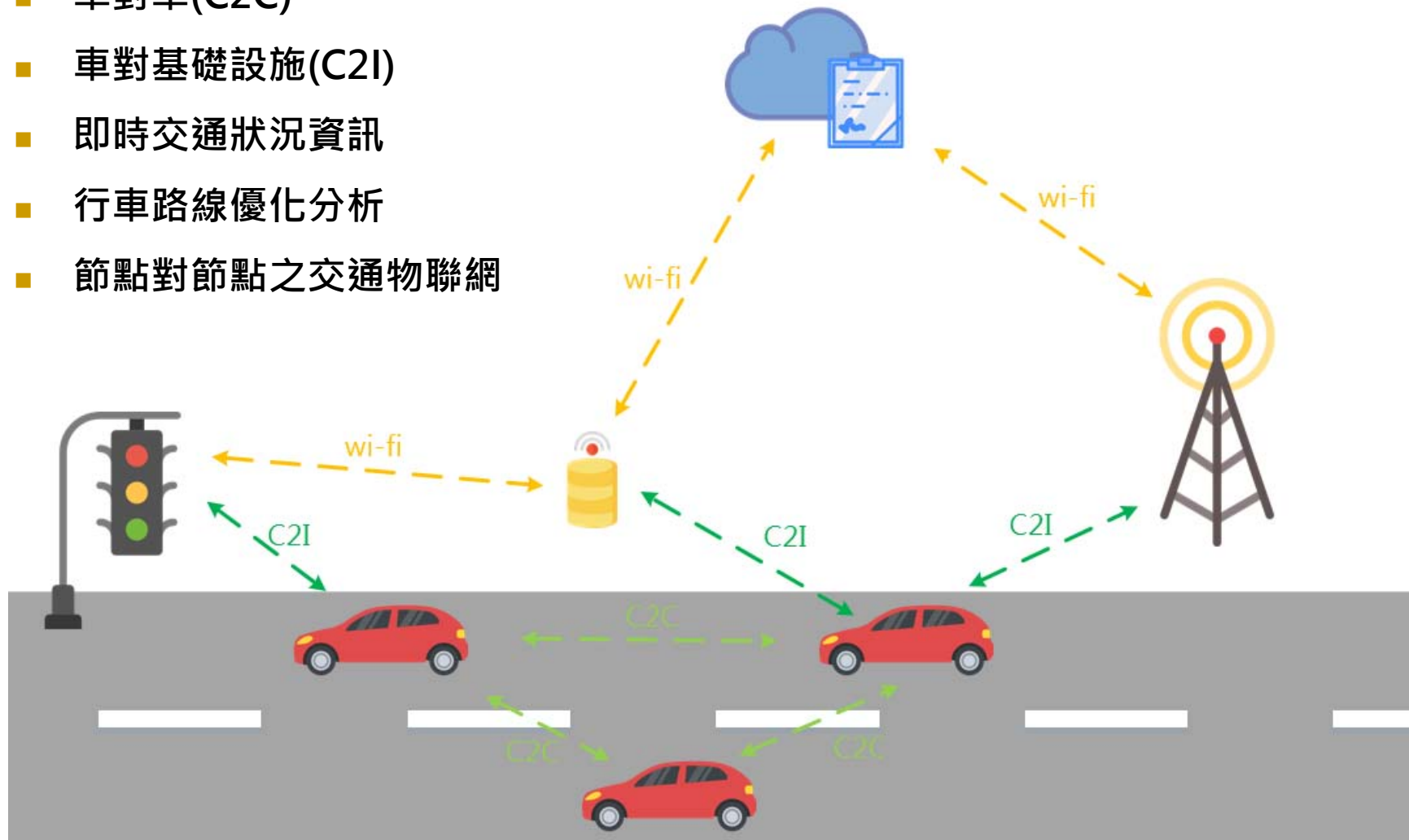
智慧農業環控

- 自動調節溫度、光照
- 自動調整施肥、灑水
- 植物種資料儲存技術
- 智慧化植物種栽培



車聯網

- 車對車(C2C)
- 車對基礎設施(C2I)
- 即時交通狀況資訊
- 行車路線優化分析
- 節點對節點之交通物聯網

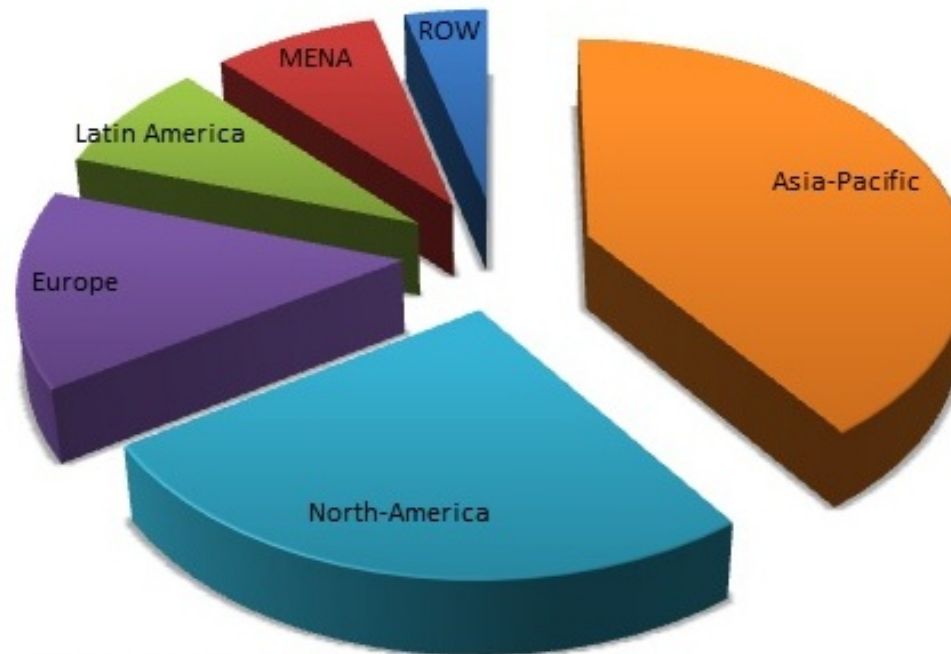




前言：物聯網市場

- 2015，全球物聯網市場達到5982億美金，其中亞洲市場佔36%
- 預估2023，全球市場將達到7242億美金，且複合年均增長率為13.2%

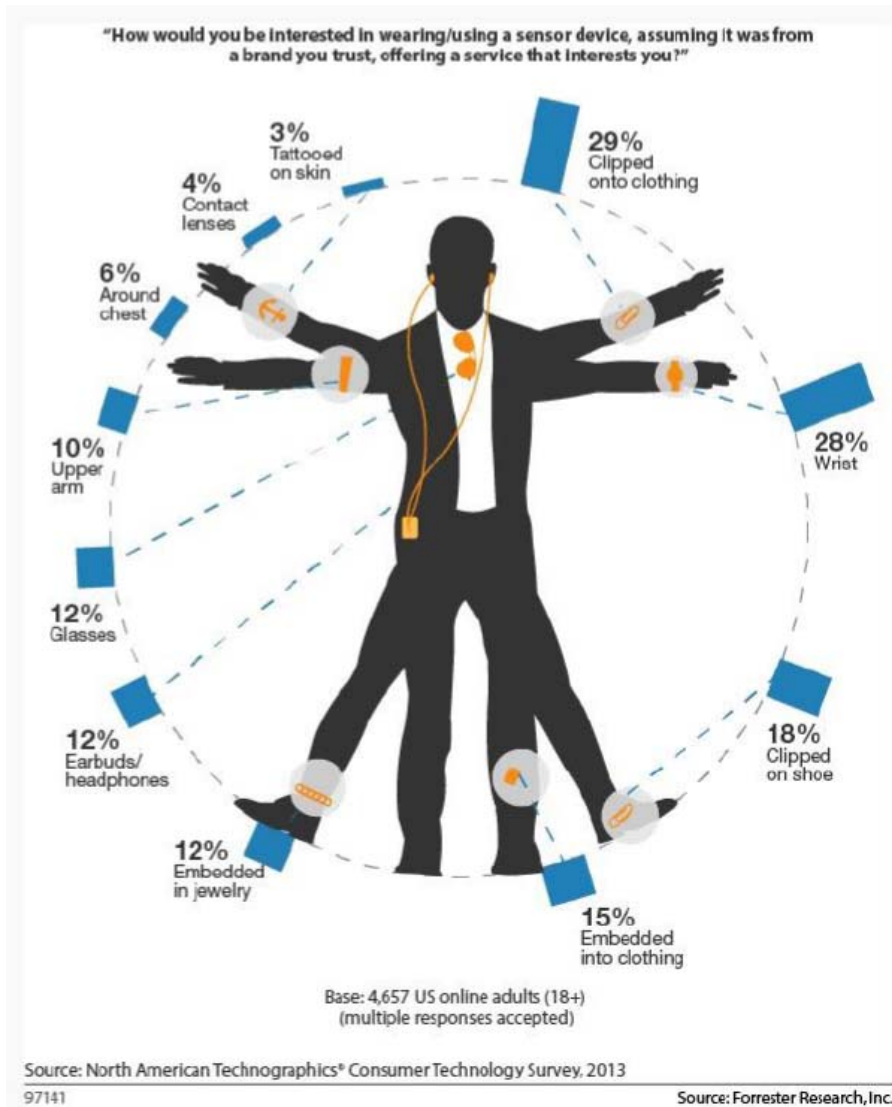
Global Internet of Things (IoT) Market Revenue Share (%), 2015



Source: Research Nester

Reference: Research Nester, Internet of Things (IoT) Market : Global Demand, Growth Analysis & Opportunity Outlook 2023

前言：生醫穿戴式產品



■ 發展背景

- 慢性病患者增加
- 健康照護意識提高
- 長照需求增加
- 智慧型手機之普及

■ 生醫穿戴式產品之功效

- 日常照護
- 健康監控
- 智慧醫療

前言：Apple Watch



- Apple Watch Series 4----ECG monitoring



Reference: Apple Watch Series 4 Keynote



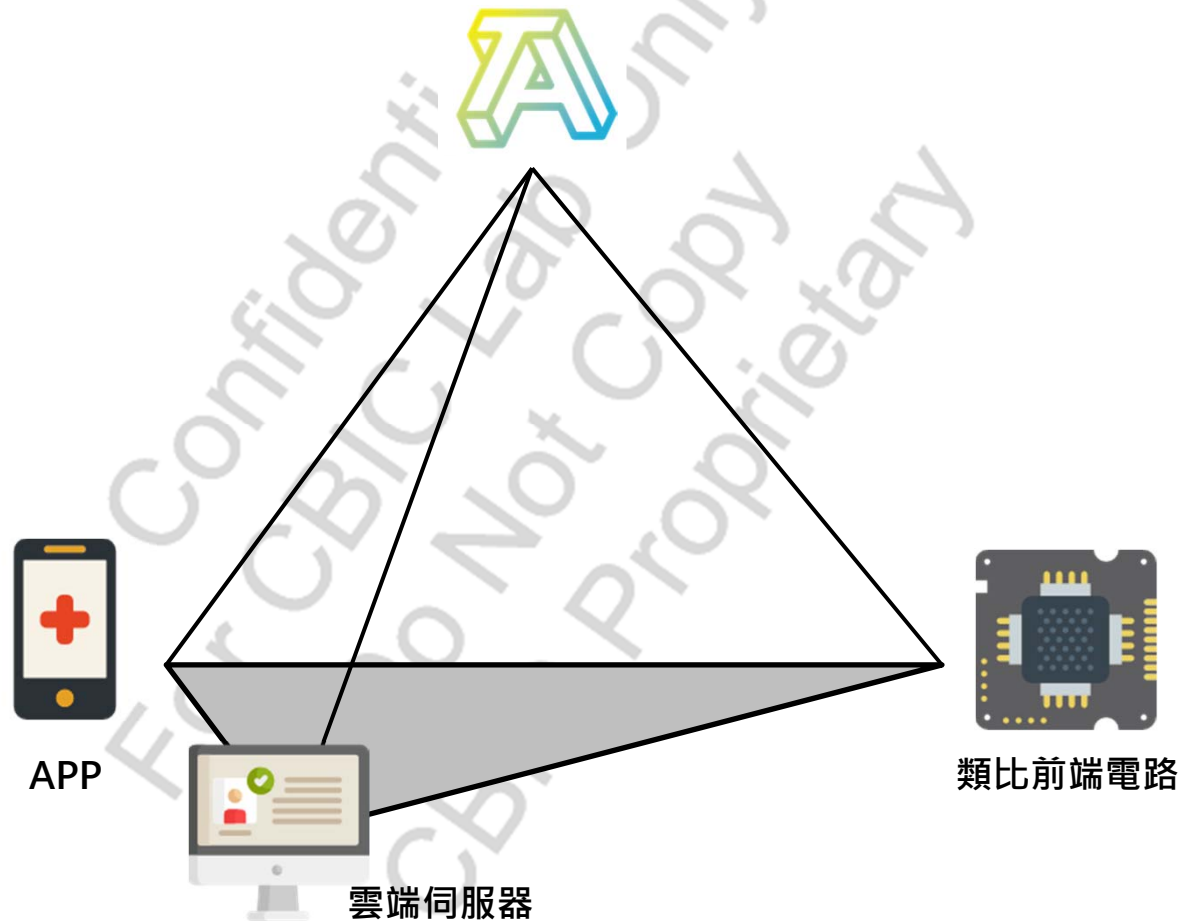
前言：Trianswer特點

- 高精確度、低功耗、微小化的生醫訊號處理模組，可量測心電、肌電等生理訊號
- 物聯網化設計，使用者可即時從APP上獲得運算結果
- 雲端大數據分析、資料庫儲存技術



前言

- Trianswer , Try and get your answer





前言：Trianswer定位



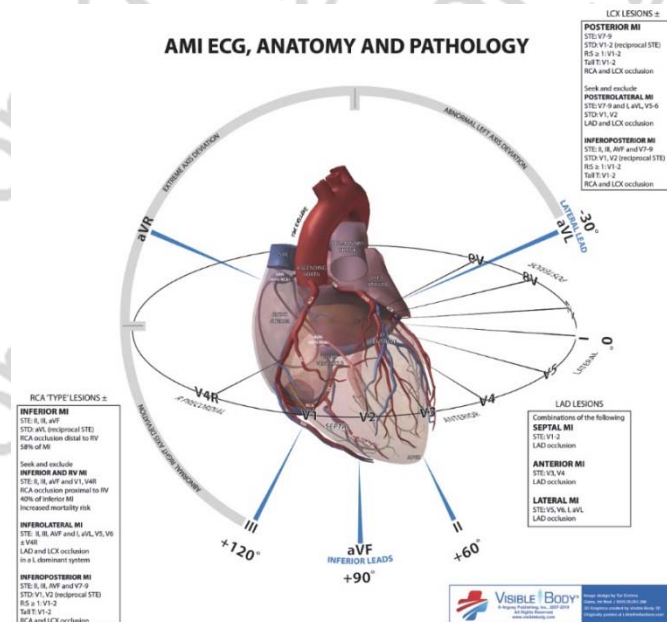


Outline

- 前言
 - 物聯網？生醫穿戴式裝置？
 - Why Trianswer？
- 生理訊號原理、Trianswer模組介紹
 - ECG(心電訊號)、EMG (肌電訊號)、PPG (光體積描述訊號)、太陽能充電、MCU (微控制器)
- Trianswer操作流程介紹

原理介紹：ECG

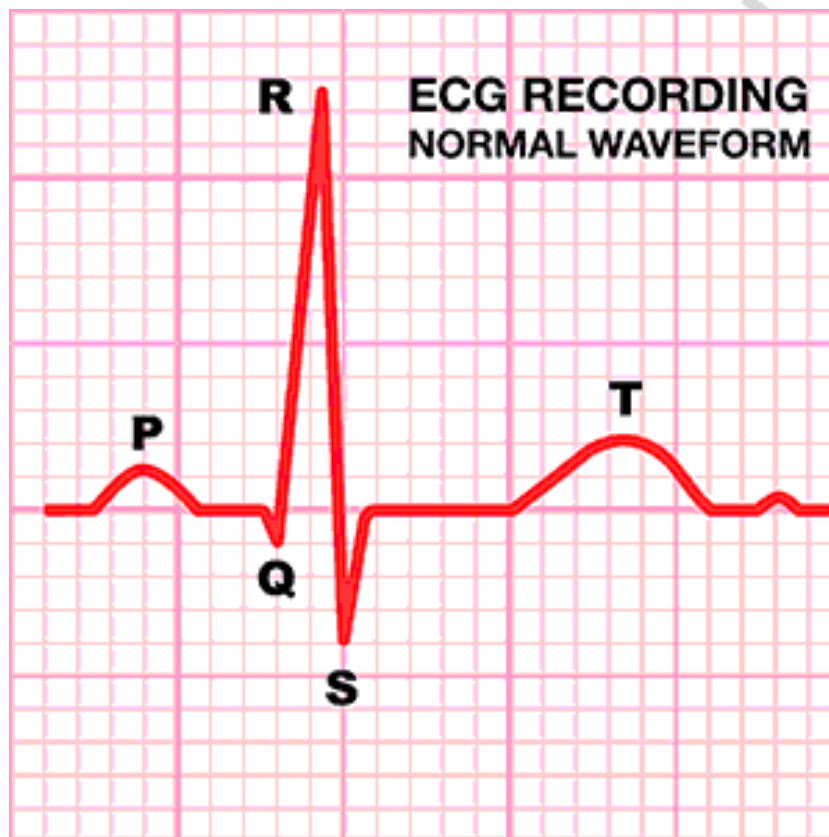
- ECG: Electrocardiography, 心電圖
 - 心臟每次跳動時會伴隨著電訊號，此電訊號即心電訊號
 - 以一組(兩個)電極，紀錄心臟的電生理活動
 - 一組電極稱做一個“導程 (Lead)”
 - 不同導程，代表以不同角度紀錄心臟的電生理活動



Reference: Visible Body

原理介紹：ECG波型解讀

■ ECG波型組成



Reference: myDr (<http://www.mydr.com.au/tests-investigations/electrocardiogram-ecg>)

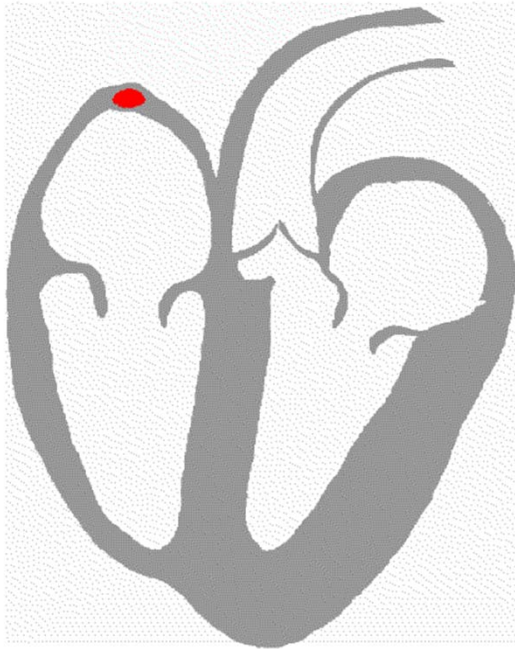
X-axis: 40ms / grid

Y-axis: 0.1mV / grid

Wave Segment	Duration Time
P wave	~80ms
P-R interval	120~200ms
QRS complex wave	80~120ms
T wave	160ms
R-R interval	0.6~1.2s (50~100bpm)



原理介紹：ECG與心臟活動關係



GIF Author: Kalumet (From Wikimedia Commons)

Atrial Excitation 心房興奮	開始	▪SA node產生自發性動作電位，並由此處開始傳出電位變化
	完成	▪電位從SA node傳到AV node ▪產生P波，代表心房產生去極化 (depolarization)
Ventricular Excitation 心室興奮	開始	▪心室開始去極化，電位往下傳導產生Q波
	完成	▪電位已傳導至整個心室，心室已完全去極化 ▪再進一步產生R波、S波
Ventricular relaxation 心室鬆弛		▪心室產生再極化，產生T波



原理介紹：ECG相關演算法

- R-peak detection: 算出心跳次數(bpm)
- HRV(Heart rate variability): 為交感、副交感神經之定量活性指標
- CNN、LSTM.....: 判別心律不整種類



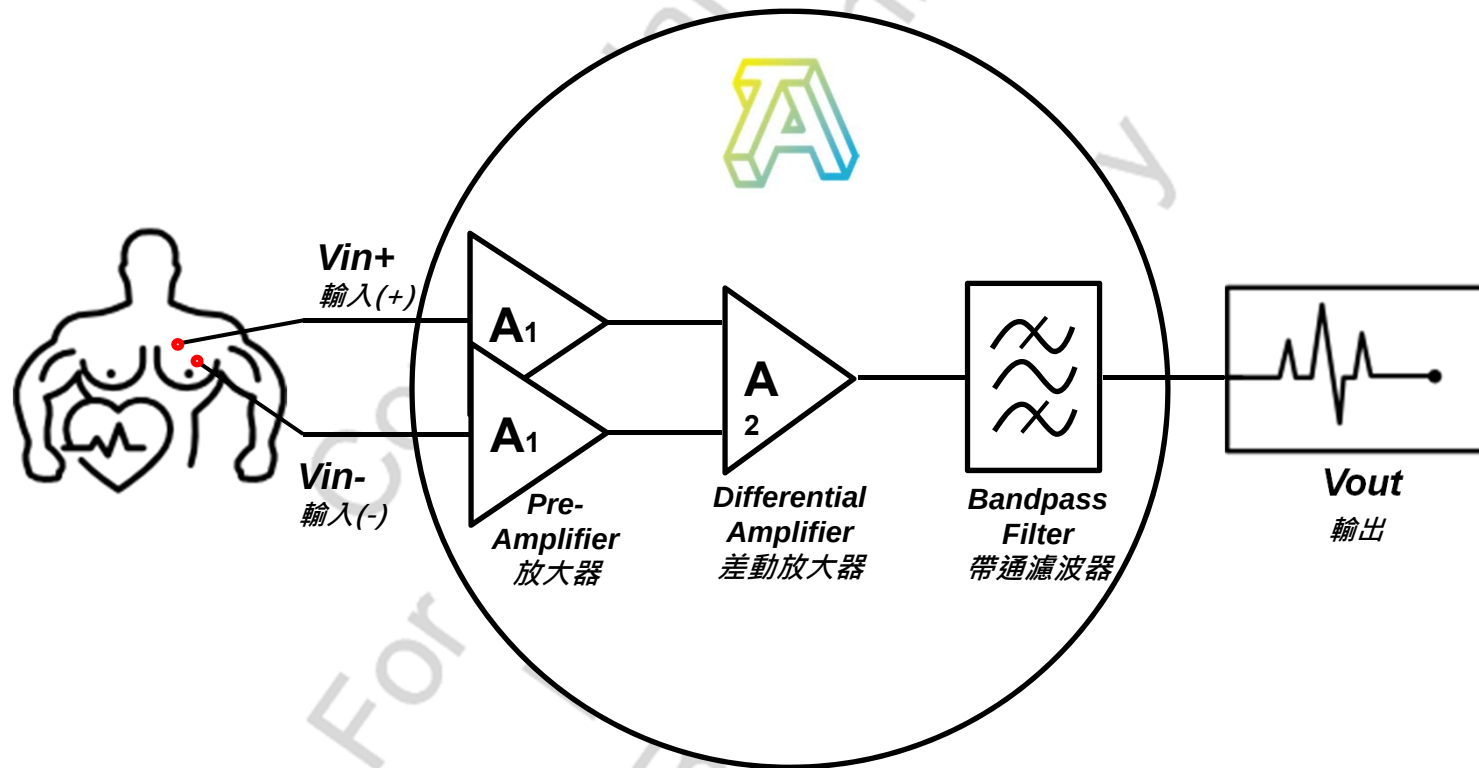
Torsades de pointes，一種多型性心室心搏過速



Third degree AV block，完全性房室傳導阻滯

模組介紹：ECG

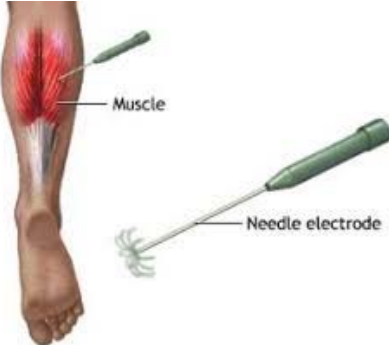
- Trianswer ECG module: 單導程量測





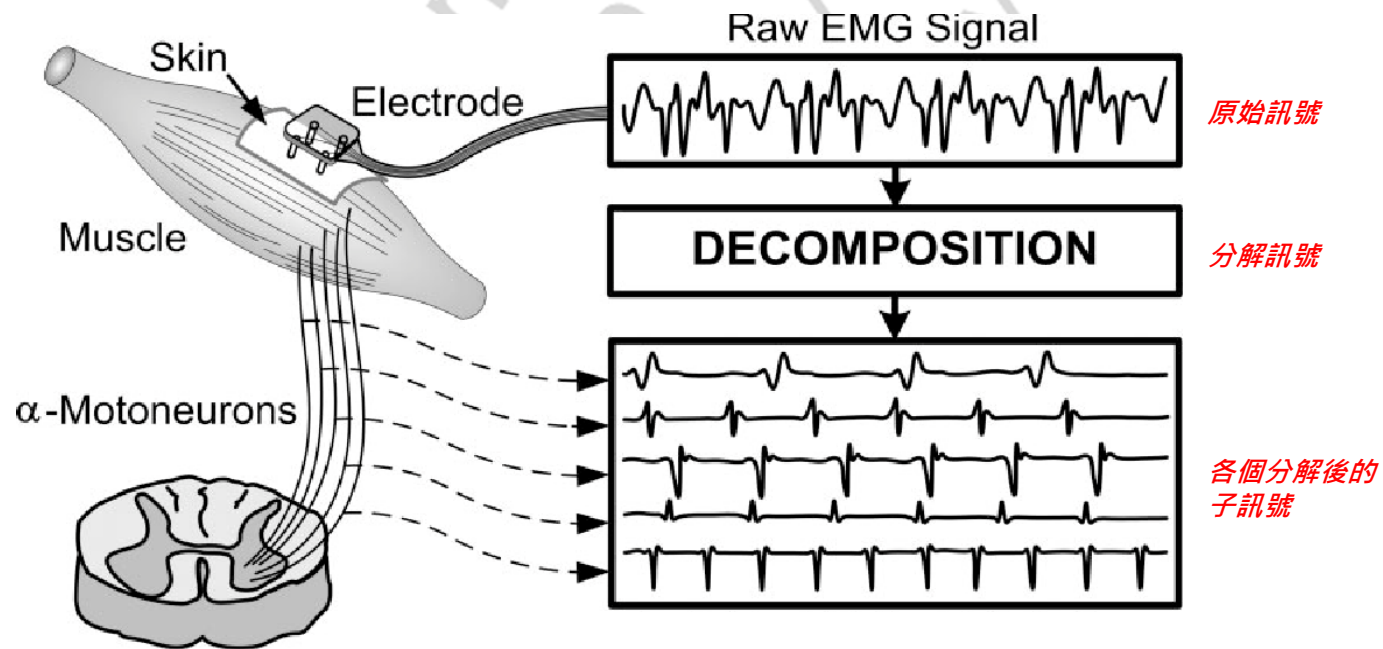
原理介紹：EMG

- EMG: Electromyography，肌電圖
 - 是一種電診斷的醫學技術，用來觀察及紀錄骨骼肌的電活動狀態
 - 量測方式：表面電極或侵入式探針

置入肌電圖(iEMG)	表面肌電圖(sEMG)
侵入式探針	表面電極
量測範圍小	量測範圍大
需由專業人員操作(消毒)	可由非專業人員操作
雜訊低	雜訊較高
 Reference: Julie N. Côté, PhD	 Reference: Biometrics Ltd.

原理介紹：EMG

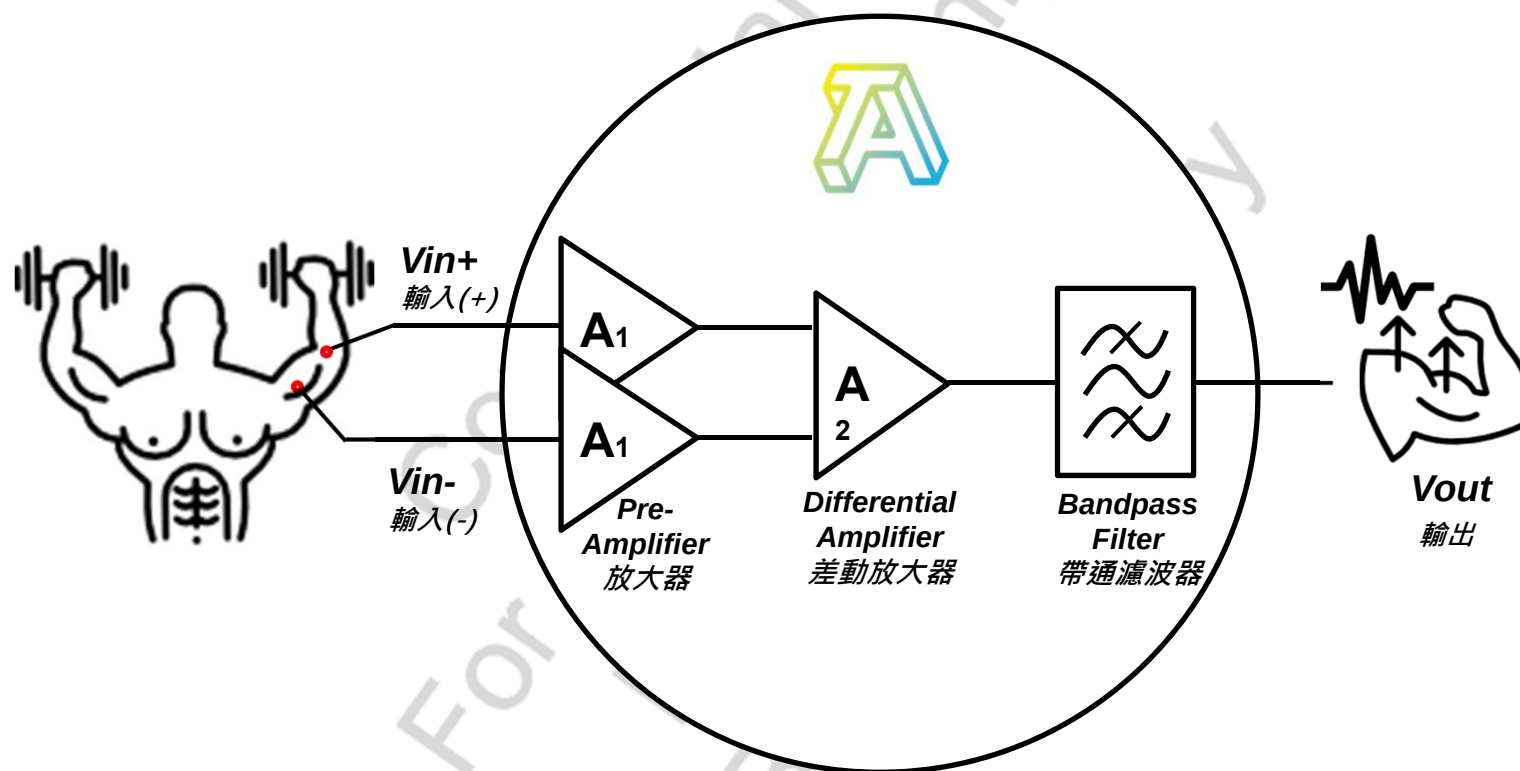
- EMG: Electromyography, 肌電圖
 - 量測結果通常為多束肌肉纖維所造成的結果，無法直觀地獲得資訊
 - 進一步分析：積分肌電圖、小波分析.....等等



Reference: Decomposition of Surface EMG Signals, Carlo J. De Luca, 2006

模組介紹：EMG

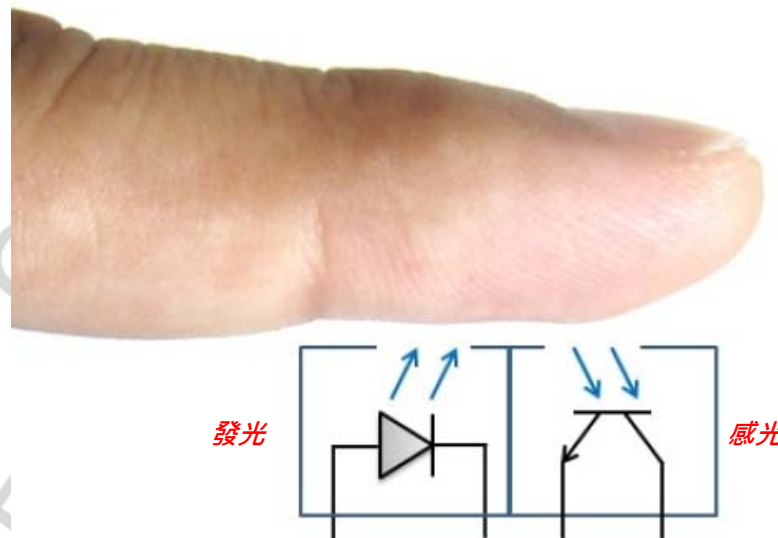
- Trianswer EMG module





原理介紹：PPG

- PPG: Photoplethysmogram，光體積變化描記法
 - 以發光二極體及光二極體做為感測元件，量測血管的收縮程度
 - 非侵入性，常應用於穿戴式裝置中，如Apple Watch、小米手環

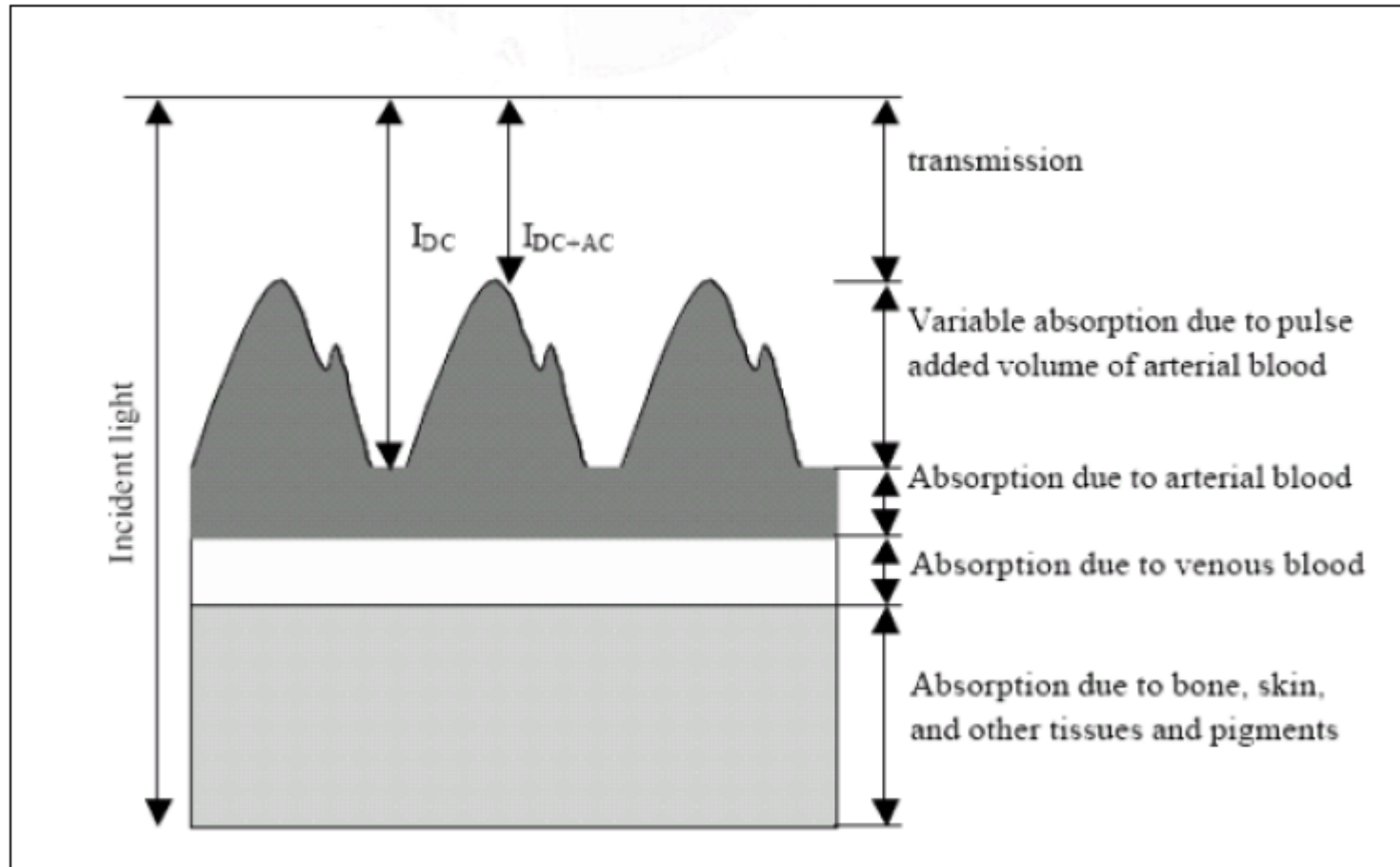


Reference: fabacademy.org



原理介紹：PPG訊號成分分析

- PPG，含有DC成分及AC成分



脈動的動脈血

不脈動的動脈血

靜脈血

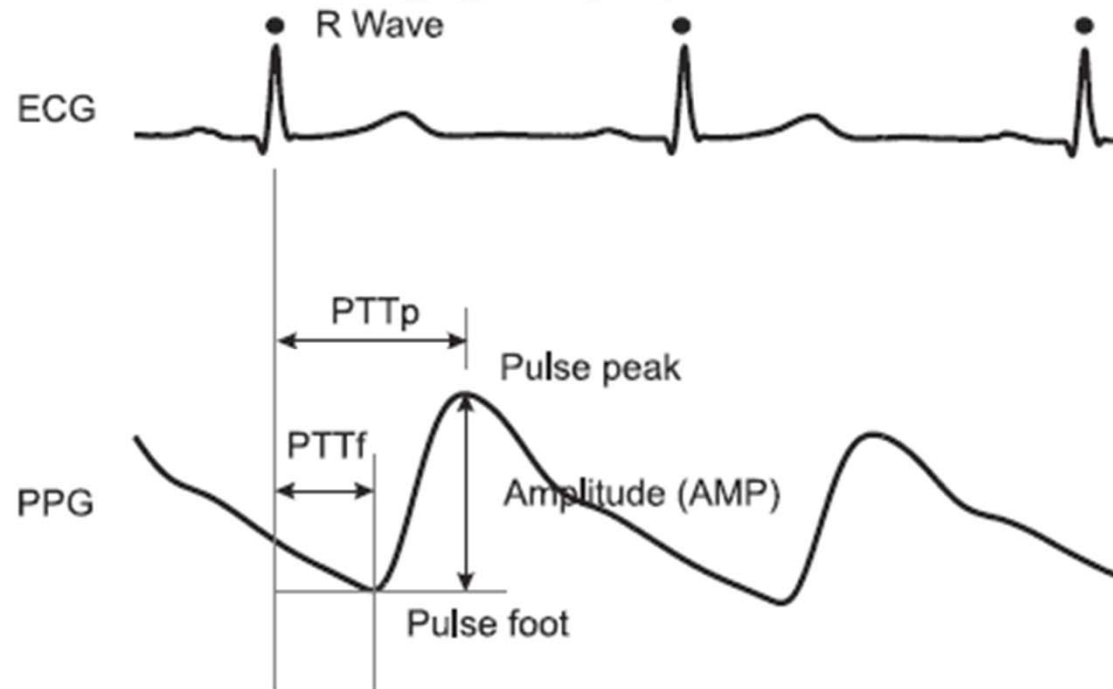
骨頭、皮膚、
組織....

Reference: Webster JG. "Design of Pulse Oximeters", IOP 1997



原理介紹：PPG訊號相關演算法

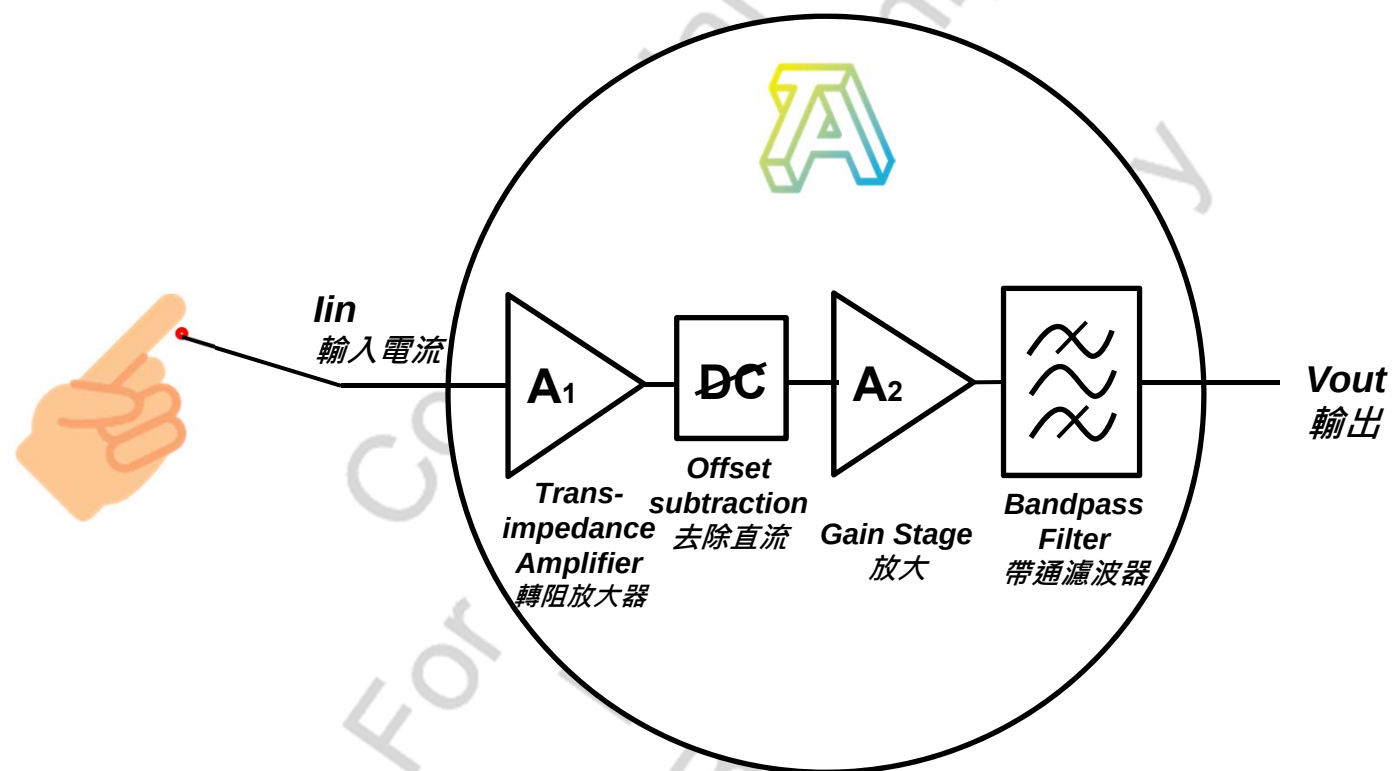
- PTT (Pulse Transit time): 搭配ECG，算出人體血壓
- 血氧飽和濃度 (SpO₂): 搭配紅光+紅外光之PPG，算出人體血氧濃度



Reference: Courtesy: John Allen; 'Photoplethysmography and its application in clinical physiological measurement.'

模組介紹：PPG

- Trianswer PPG module





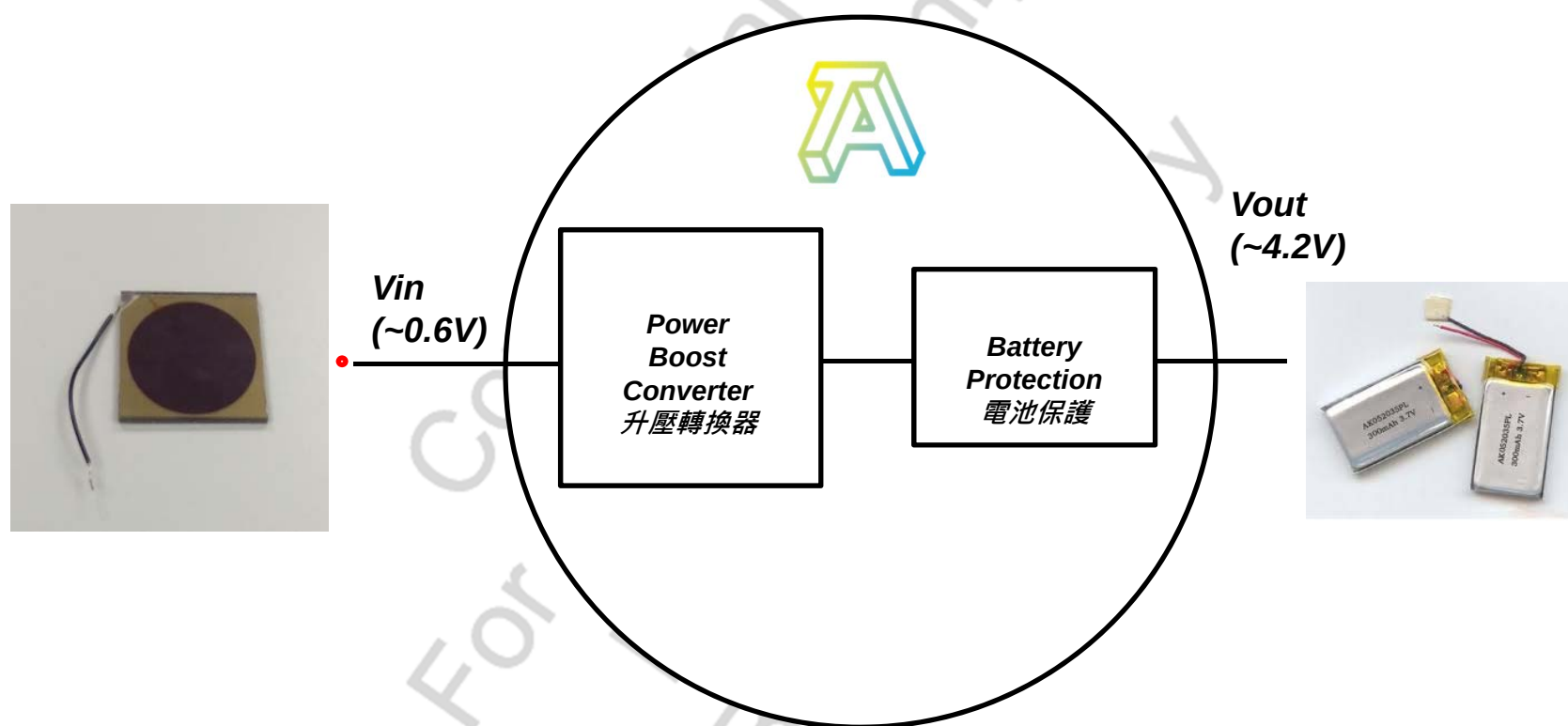
原理介紹：太陽能充電

- 將Energy harvesting(獵能)搭配低功耗之模組及傳輸方式
 - 延長穿戴式裝置之使用時間
- 太陽能板採用中興大學所研發的染敏太陽能電池
 - 特點: 在室內獲取**室內日光燈源**即可使用



模組介紹：PPG

- Trianswer Solar power module



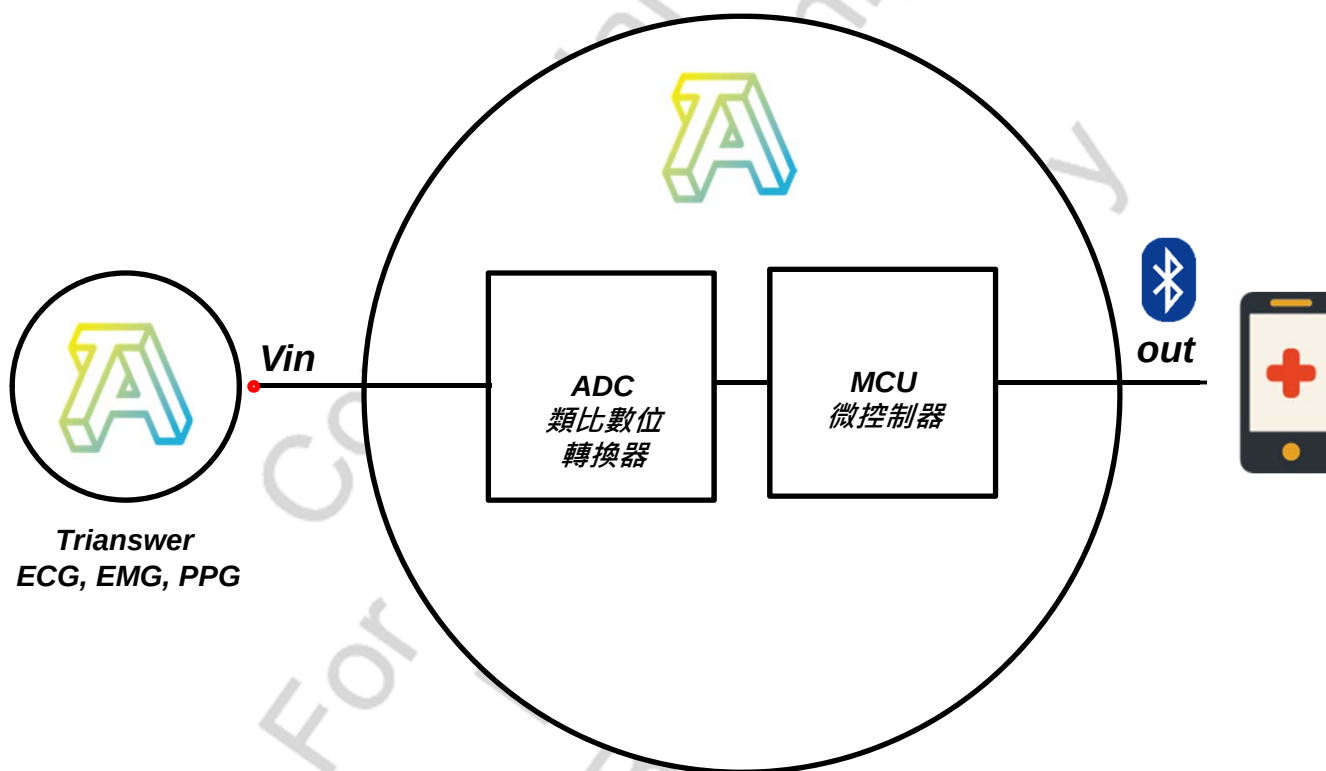


原理介紹：MCU模組

- MCU，Micro control unit，微控制器，相較於電腦上面的CPU體積較小
 - 更容易存在於穿戴式產品中
- 常見的MCU
 - Atmel系列(下一個課程會帶大家認識)
 - ARM Cortex-M0、M3、M4系列
- 藍芽傳輸: BLE5.0

模組介紹：MCU

- Trianswer MCU module



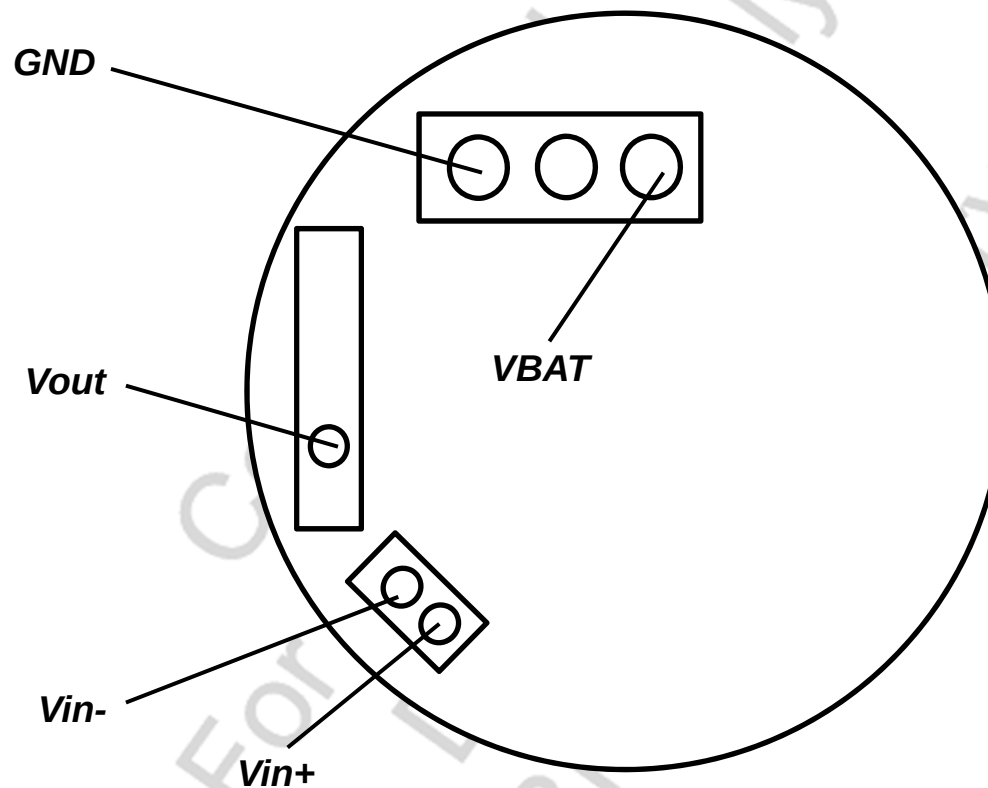


Outline

- 前言
 - 物聯網？生醫穿戴式裝置？
 - Why Trianswer？
- 生理訊號原理、Trianswer模組介紹
 - ECG(心電訊號)、EMG(肌電訊號)、PPG(光體積描述訊號)、太陽能充電、MCU(微控制器)
- Trianswer操作流程介紹

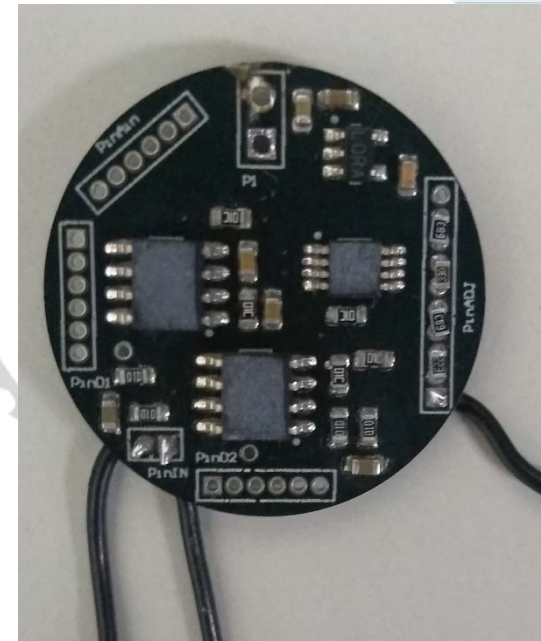
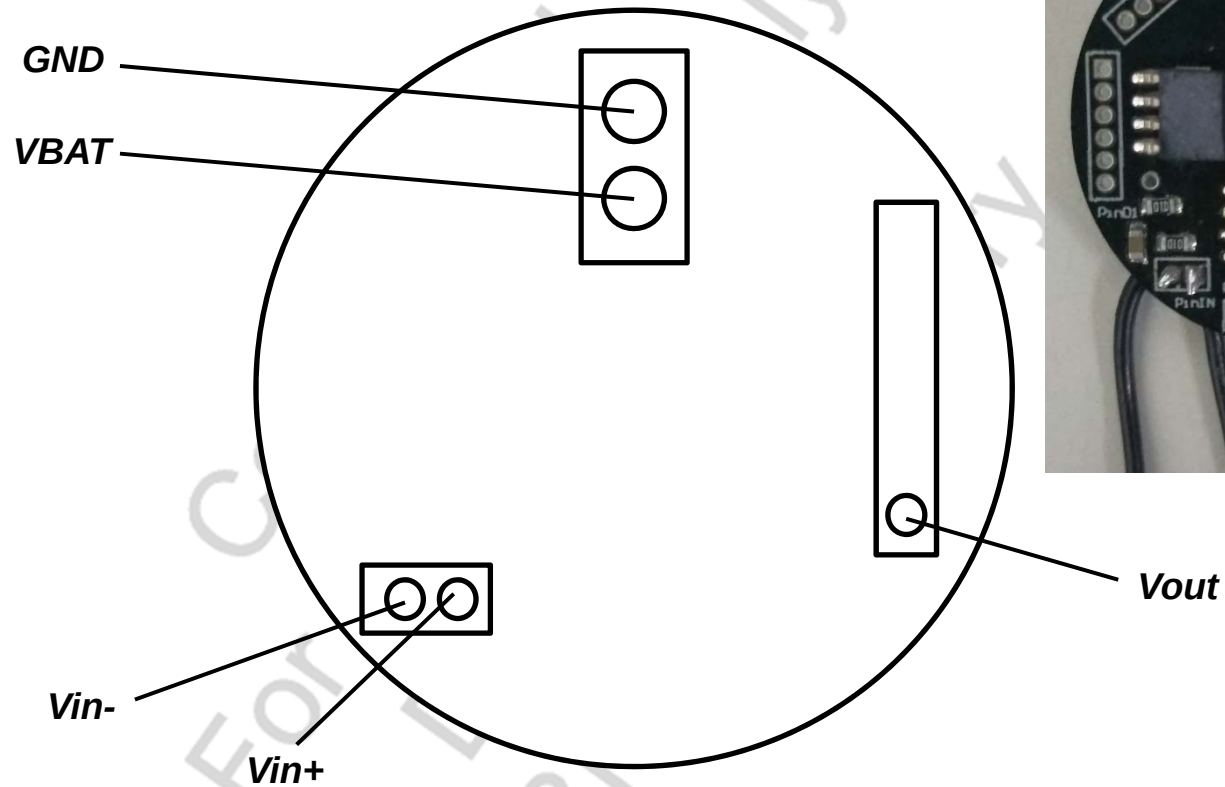
操作介紹：ECG

- Trianswer ECG module



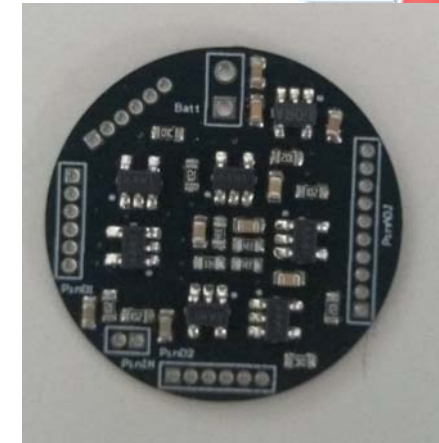
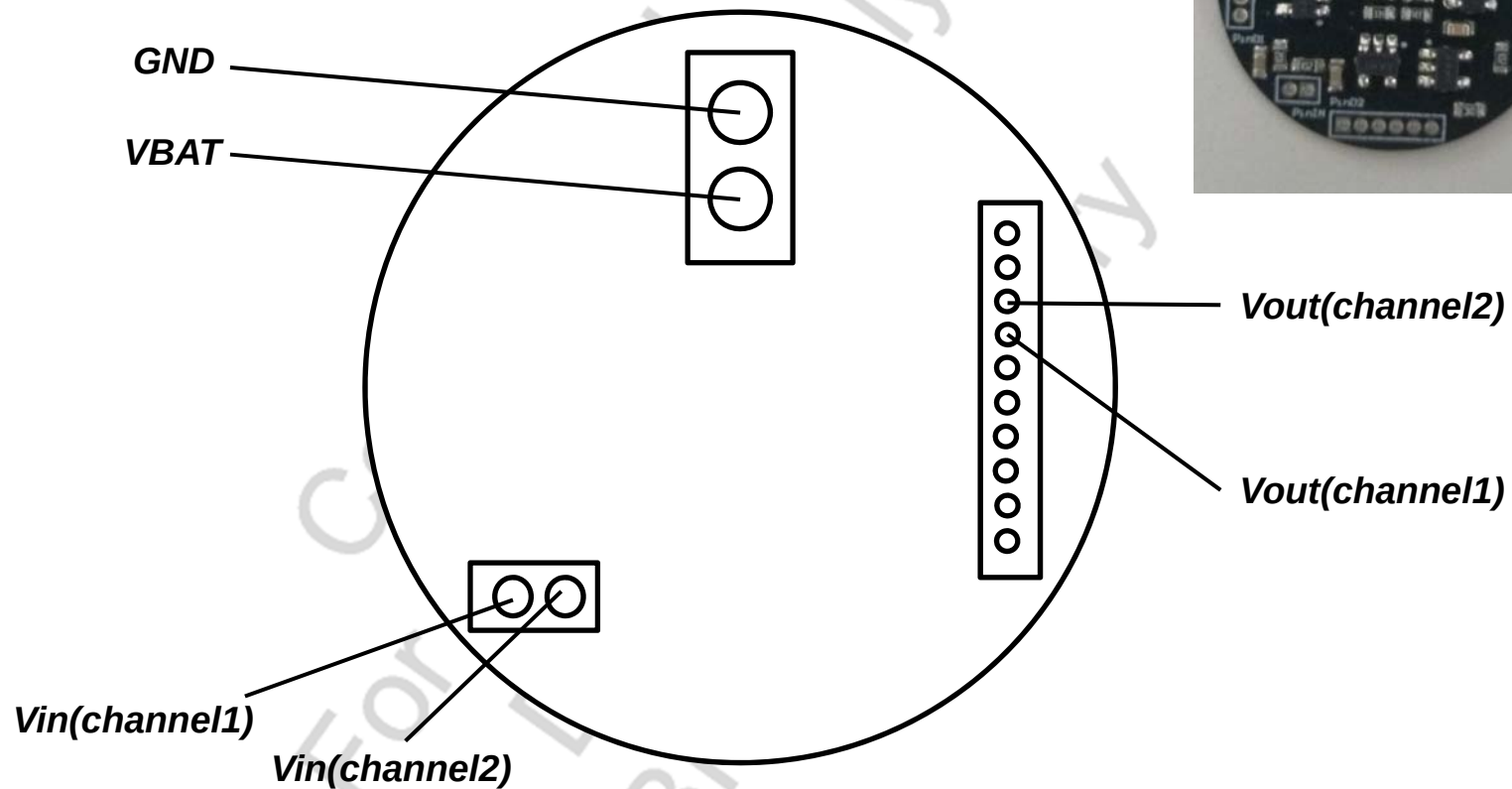
操作介紹：EMG

- Trianswer EMG module



操作介紹：PPG

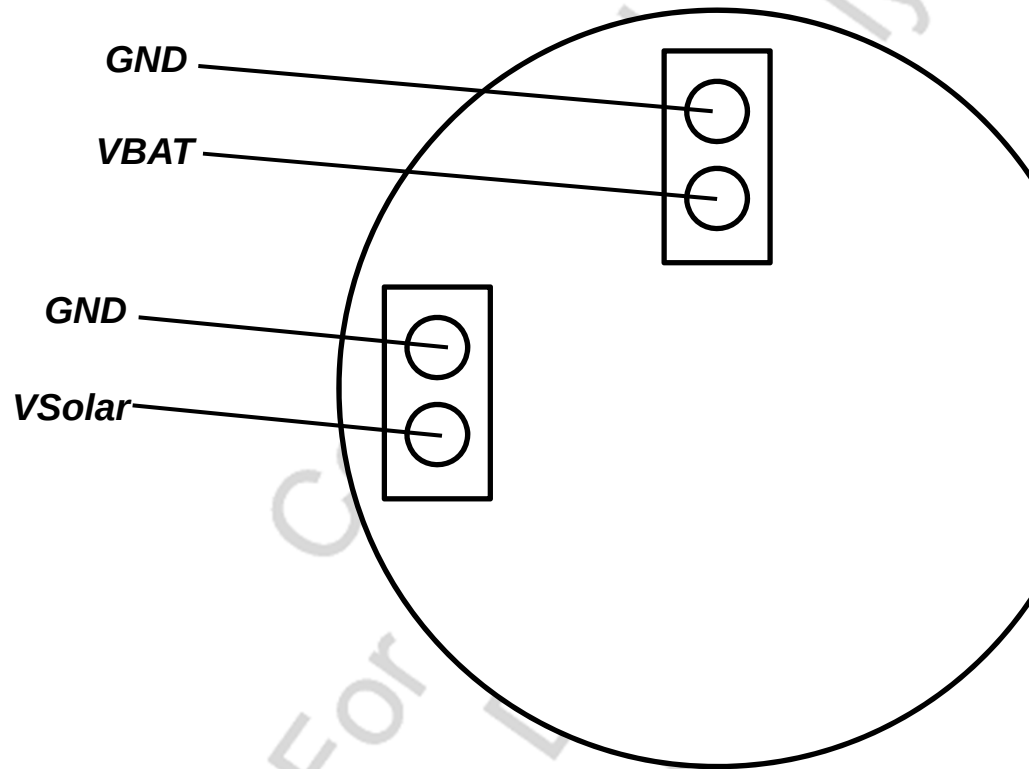
- Trianswer PPG module





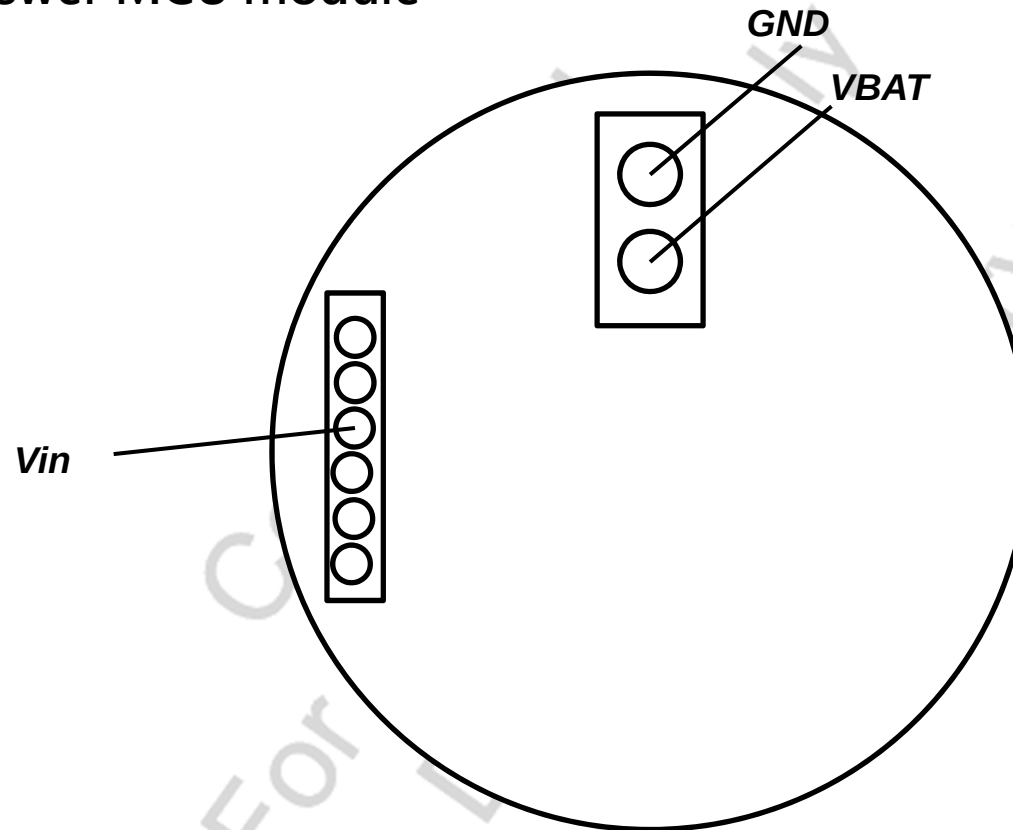
操作介紹：太陽能充電

- Trianswer Solar power module



操作介紹：MCU

- Trianswer MCU module





操作注意事項

- 每個模組的輸入電源請勿超過16V，建議電源可用3.7V鋰電池；9V電池亦可，端看使用者應用情景。
- 使用ECG模組時，如果在Trianswer APP上看到類似上下相反的ECG波型時，請對調“Vin+”及“Vin-”這兩個腳位的輸入。
- MCU模組：
 - 此藍芽模組僅可一對一連接，如需更換連接裝置，請先取消原智慧型裝置之連接，再進行更換。



Thanks!!

生醫穿戴式裝置開發課程模組

課程: 用Arduino算出心率

Presenter : Tony助教

Date : 2020/01/30~2021/12/31





Outline

- 前言
- 設計概念
- 實作
- 總結

Confidential
For CBIC Lab Only
Do Not Copy
CBIC Proprietary



前言 – Arduino ?

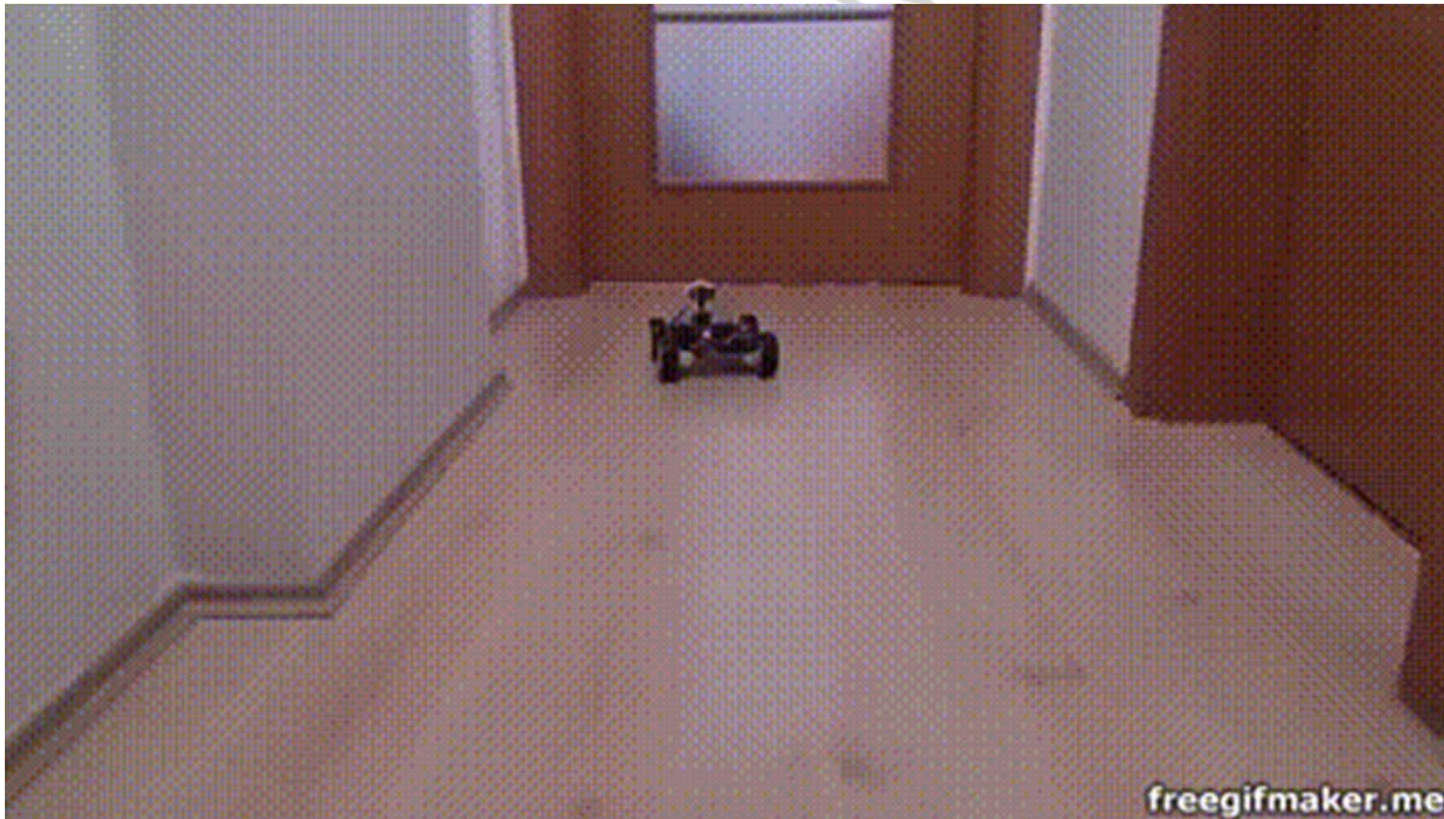


- Arduino：一款“開源”的“嵌入式系統”開發平台
- “開源”：開放的硬體電路設計圖及軟體原始碼
 - <https://opensource.com/tags/arduino>
- “嵌入式系統”：完成特定任務的系統，不是解決普遍任務的
 - 核心：在微處理機(Micro Control Unit, MCU)上面預先編程好的幾個任務
 - 通常嵌入式系統上的軟體是不太變動的，也稱作“韌體”
 - 日常生活中的嵌入式系統：冰箱控溫、防鎖死煞車系統.....等等
- 可以搭載許多不同的擴充板(shield)或感測器，增加其他功能
 - 超聲波、溫度、加速度計、乙太網路.....等等

前言 – Arduino能開發什麼？



- 超聲波避障車



Source: <https://www.youtube.com/watch?v=C3GhJmmRY4Y>



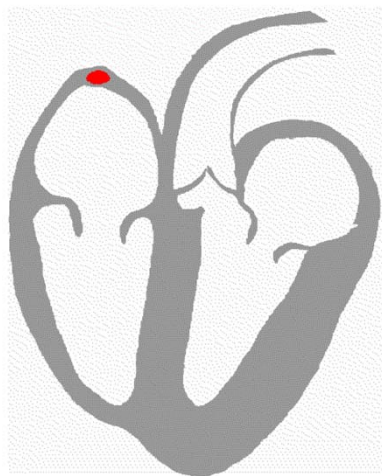
Outline

- 前言
- 設計概念
 - 心率的計算
 - Arduino的語言
- 實作
- 總結

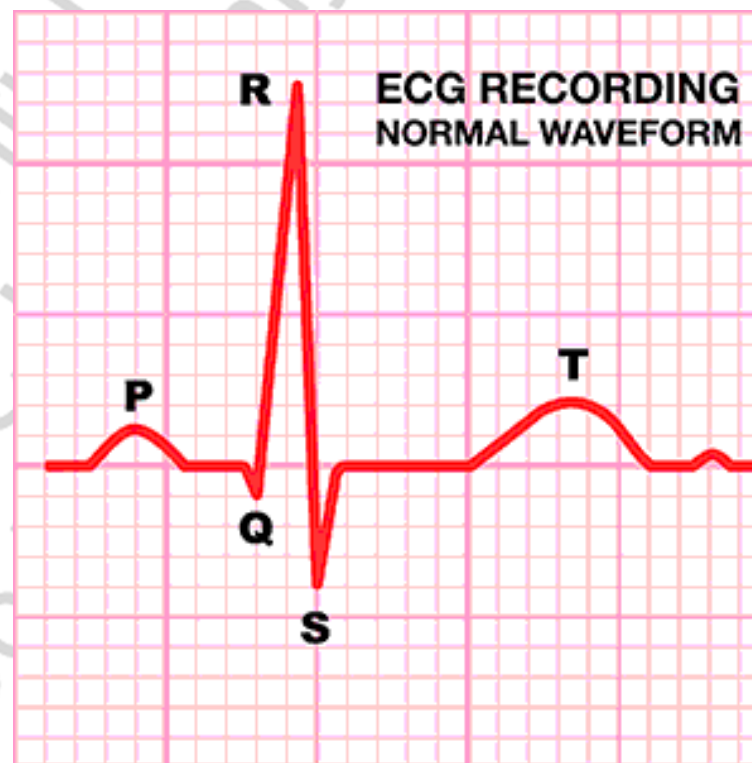
Confidential
For CBIC Lab Only
Do Not Copy
CBIC Proprietary

ECG波型特色

■ 心電訊號：ECG



GIF Author: Kalumet (From Wikimedia Commons)



Reference: myDr (<http://www.mydr.com.au/tests-investigations/electrocardiogram-ecg>)

Y-axis: 0.1mV / grid

X-axis: 40ms / grid



演算法 – 心率計算

- Beats per minute (bpm)
- 心率可以得知是否有不正常的心臟活動現象
 - 過速心搏(Tachycardias) : $>100\text{bpm}$
 - 過緩心搏(Bradycardias) : $<60\text{bpm}$



Torsades de pointes · 一種多型性心室心搏過速

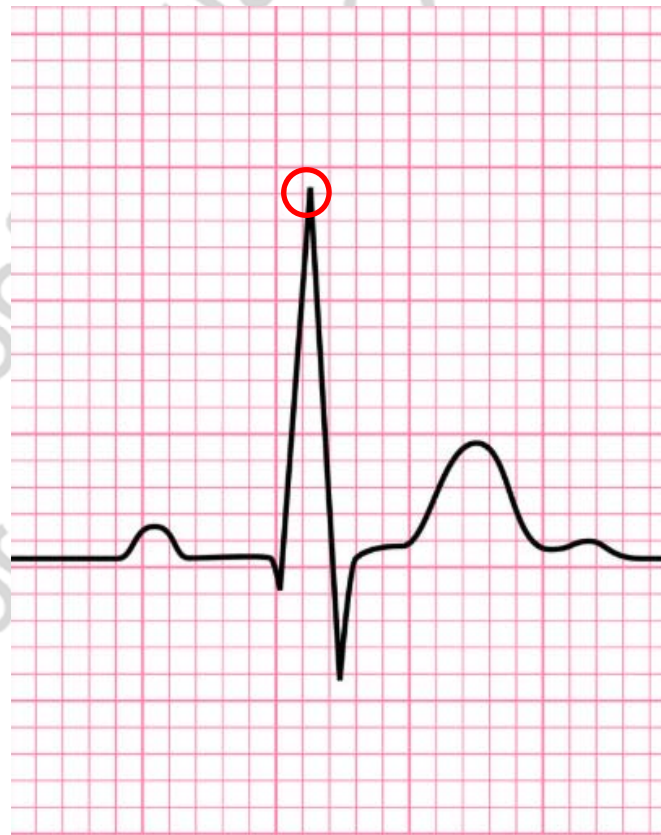


Third degree AV block · 完全性房室傳導阻滯



Tony-Kevin detection

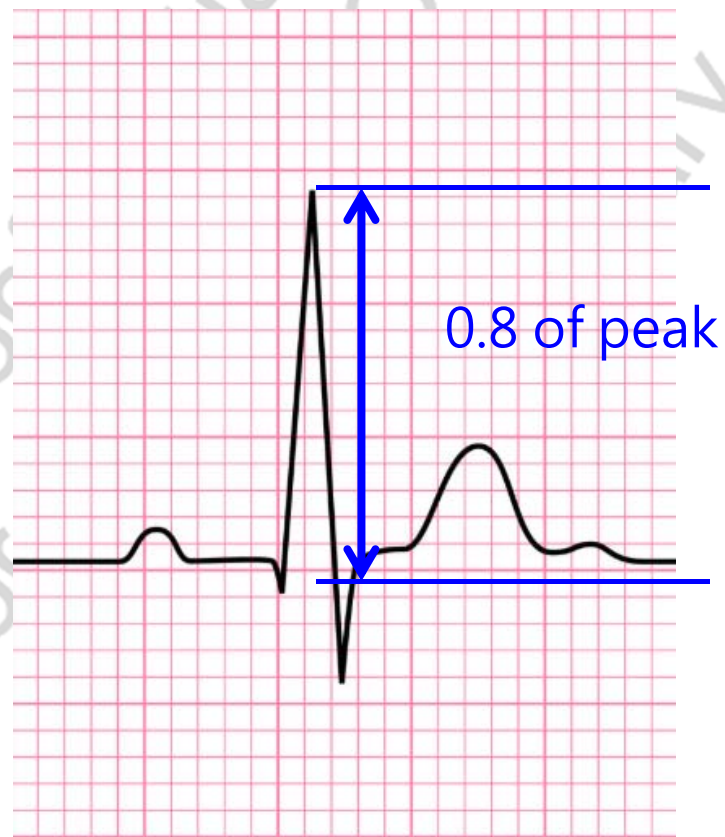
- 區域極大值+閾值偵測
- Step 1: 找到區域最大值





Tony-Kevin detection(cont.)

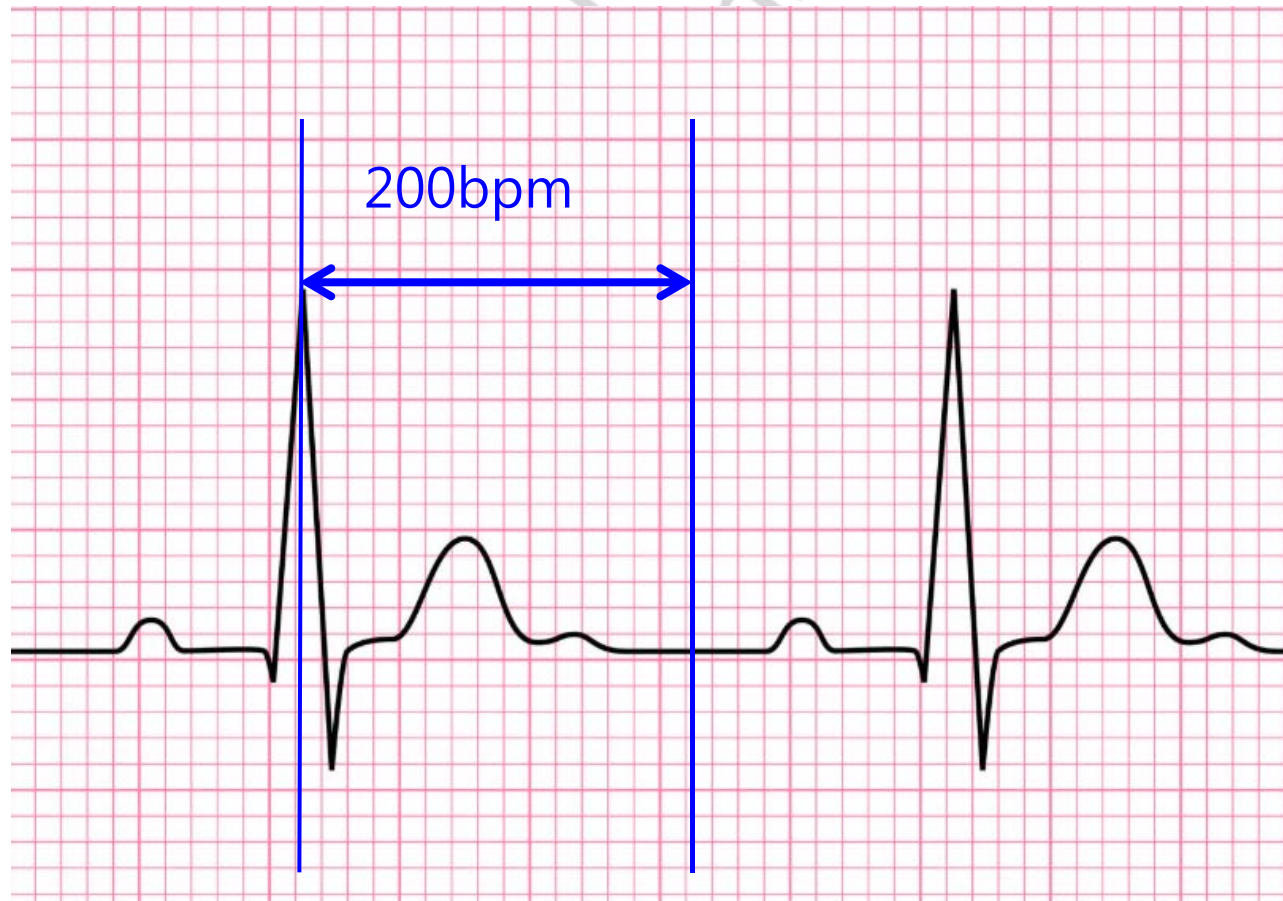
- Step 2: 這個最大值是否有超過所設定的閾值呢？
 - 目的：避免將P波、T波誤判成R peak





Tony-Kevin detection(cont.)

- Step 3: 確定兩個R peak間隔是否有超過200bpm ?
 - 目的：避免將移動所造成的雜訊或者是很大的T波誤判成R peak





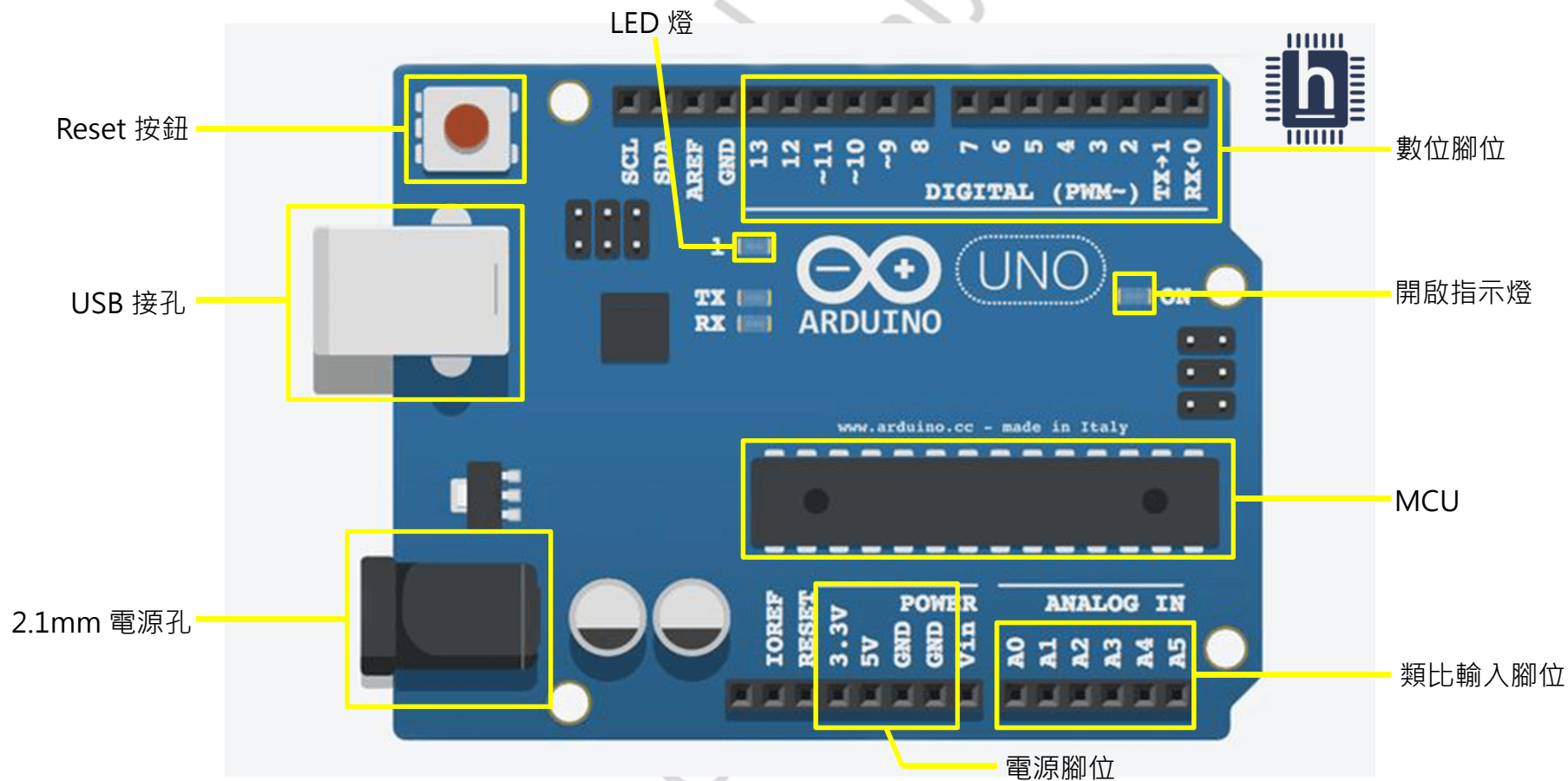
Tony-Kevin detection(cont.)

- Step 4: 算出R-R interval後，反推出bpm
 - $\text{bpm} = (1/\text{R-R interval}) * 60$
 - 每次新的peak被偵測到，就可以更新一次bpm
 - 優點：更新速度很快
 - 缺點：平均起來的準確度可能相較於5秒平均一次來得低
 - Case 1: $1.2 \text{ beats/sec} \Rightarrow 72 \text{ bpm}$
 - Case 2: $7 \text{ beats/5sec} \Rightarrow 84 \text{ bpm}$



Arduino的外觀

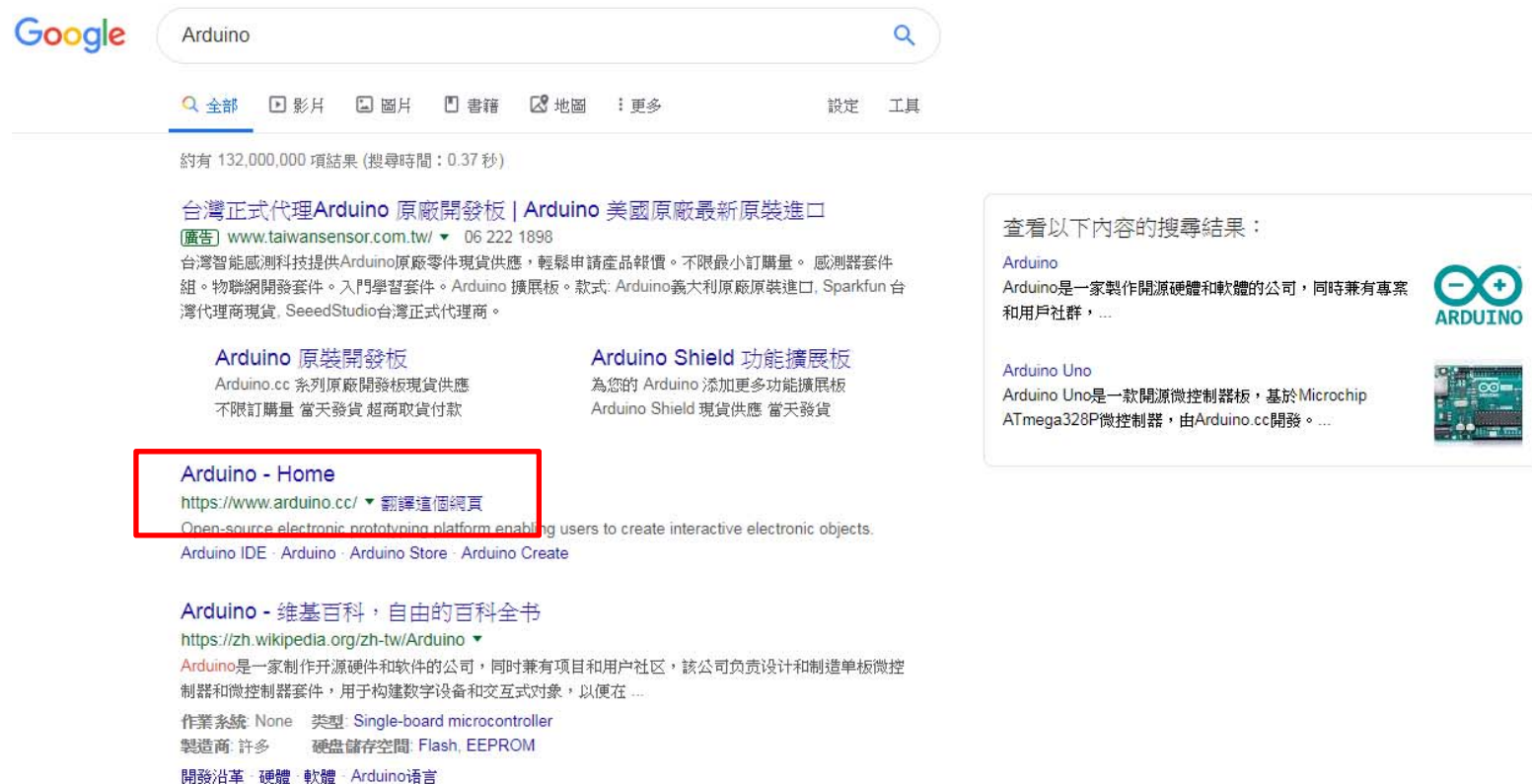
■ Arduino外觀介紹 (Uno)



Source: <https://www.hackerearth.com/blog/internet-of-things/a-tour-of-the-arduino-uno-board/>

安裝Arduino IDE(1)

- 打開Google，搜尋“ Arduino ”，點進Arduino官網



The screenshot shows a Google search for "Arduino". The search bar at the top contains the word "Arduino". Below the search bar, there are tabs for "全部" (All), "影片" (Videos), "圖片" (Images), "書籍" (Books), "地圖" (Maps), and "更多" (More). The search results show approximately 132,000,000 results in 0.37 seconds. The first result is "台灣正式代理Arduino 原廠開發板 | Arduino 美國原廠最新原裝進口" with the URL "www.taiwansensor.com.tw/". Below this, there are two columns of results: "Arduino 原廠開發板" and "Arduino Shield 功能擴展板". A red box highlights the "Arduino - Home" result with the URL "https://www.arduino.cc/". To the right of the search results, there is a sidebar with the heading "查看以下內容的搜尋結果：" (View search results for the following content:). It includes a section for "Arduino" with a description and the Arduino logo, and a section for "Arduino Uno" with a description and an image of the Arduino Uno board.

Google 搜尋 Arduino

約有 132,000,000 項結果 (搜尋時間: 0.37 秒)

台灣正式代理Arduino 原廠開發板 | Arduino 美國原廠最新原裝進口
(廣告) www.taiwansensor.com.tw/ 06 222 1898
台灣智能感測科技提供Arduino原廠零件現貨供應，輕鬆申請產品報價。不限最小訂購量。感測器套件組。物聯網開發套件。入門學習套件。Arduino 擴展板。款式: Arduino義大利原廠原裝進口, Sparkfun 台灣代理商現貨, SeeedStudio台灣正式代理商。

Arduino 原廠開發板
Arduino.cc 系列原廠開發板現貨供應
不限訂購量 當天發貨 超商取貨付款

Arduino Shield 功能擴展板
為您的 Arduino 添加更多功能擴展板
Arduino Shield 現貨供應 當天發貨

Arduino - Home
https://www.arduino.cc/ 翻譯這個網頁
Open-source electronic prototyping platform enabling users to create interactive electronic objects.
Arduino IDE · Arduino · Arduino Store · Arduino Create

Arduino - 維基百科，自由的百科全书
https://zh.wikipedia.org/zh-tw/Arduino
Arduino是一家制作开源硬件和软件的公司，同时兼有项目和用户社区，该公司负责设计和制造单板微控制器和微控制器套件，用于构建数字设备和交互式对象，以便在 ...
作業系統: None 類型: Single-board microcontroller
製造商: 許多 硬盤儲存空間: Flash, EEPROM
開發沿革 硬體 軟體 Arduino语言

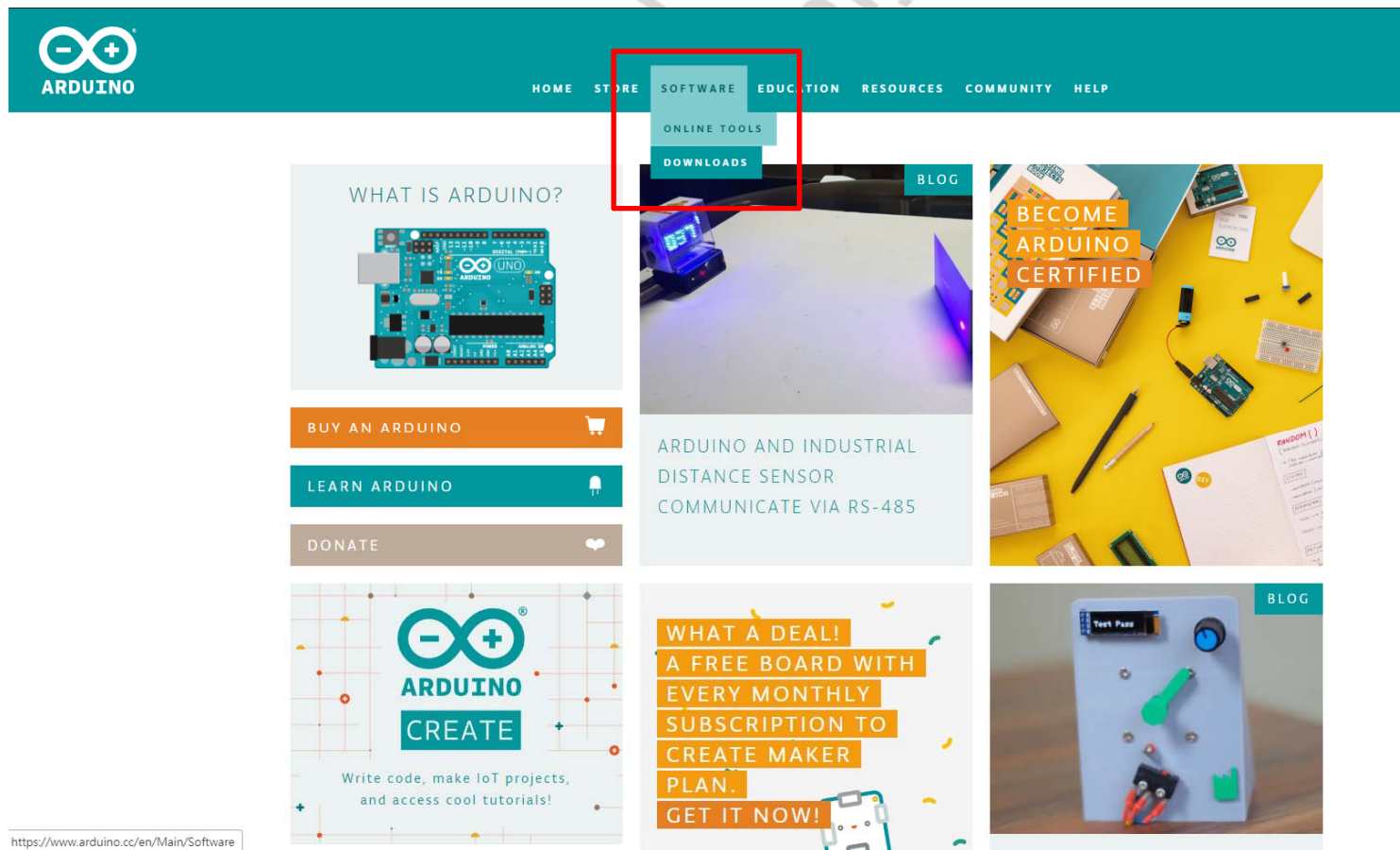
查看以下內容的搜尋結果：

Arduino
Arduino是一家製作開源硬體和軟體的公司，同時兼有專案和用戶社群，...

Arduino Uno
Arduino Uno是一款開源微控制器板，基於Microchip ATmega328P微控制器，由Arduino.cc開發。...

安裝Arduino IDE(2)

- 點 “SOFTWARE” 按鈕，接著點 “DOWNLOADS” 按鈕



安裝Arduino IDE(3)

- 下載Arduino的IDE，並按照自己電腦的版本下載安裝檔

The screenshot shows the Arduino IDE download page. A red box highlights the text "Download the Arduino IDE". Another red box highlights the "Windows Installer, for Windows XP and up" link. A third red box highlights the "Windows app" link, which includes the text "Requires Win 8.1 or 10" and a "Get" button. The page also features the Arduino logo, version information (1.8.9), and links for "HOURLY BUILDS" and "BETA BUILDS".

Download the Arduino IDE

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10
Get

Mac OS X 10.8 Mountain Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Release Notes
Source Code
Checksums (sha512)

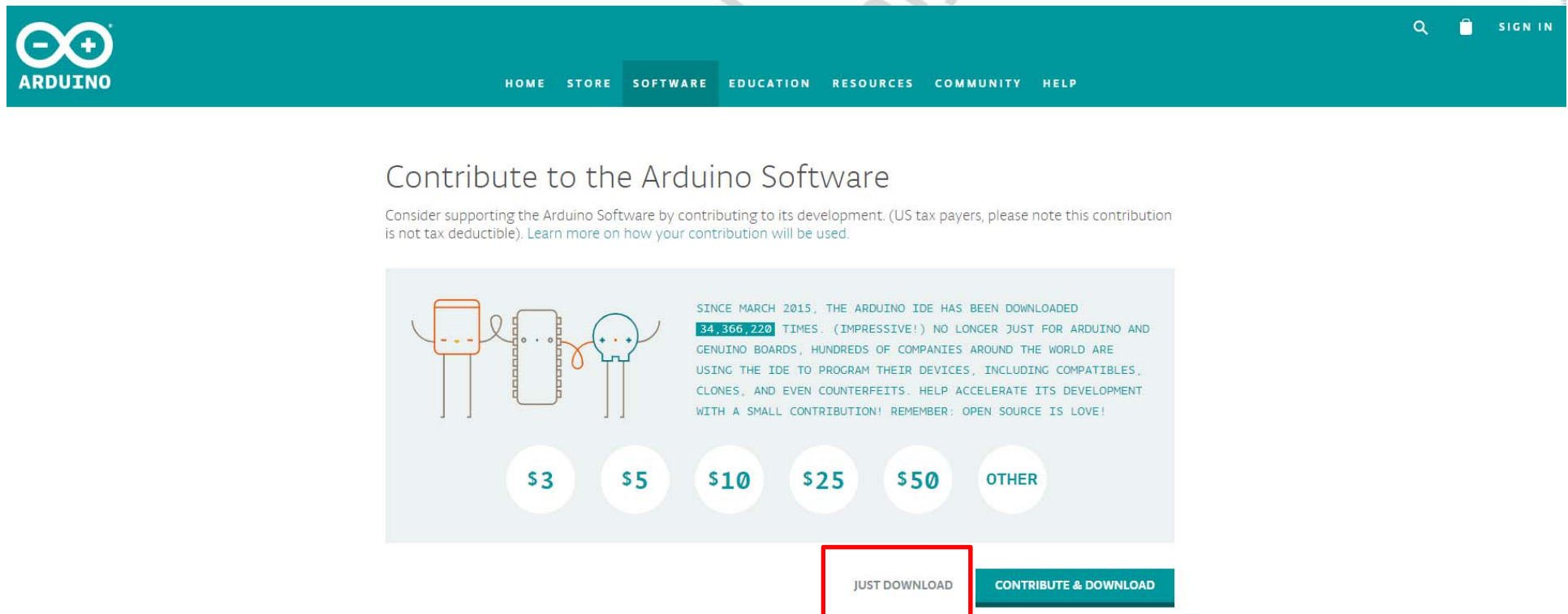
ARDUINO 1.8.9
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

HOURLY BUILDS
Download a **preview of the incoming release** with the most updated features and bugfixes.
Windows

BETA BUILDS
Download the **Beta Version** of the Arduino IDE with experimental features. This version should NOT be used in production.
Windows

安裝Arduino IDE(4)

- 按“ JUST DOWNLOAD” ，並下載安裝檔到桌面

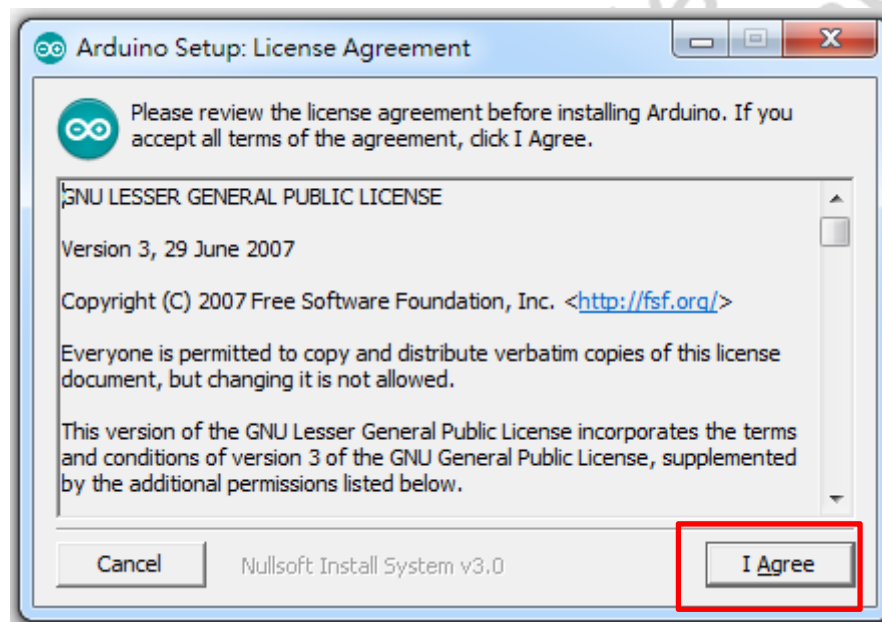


安裝Arduino IDE(5)

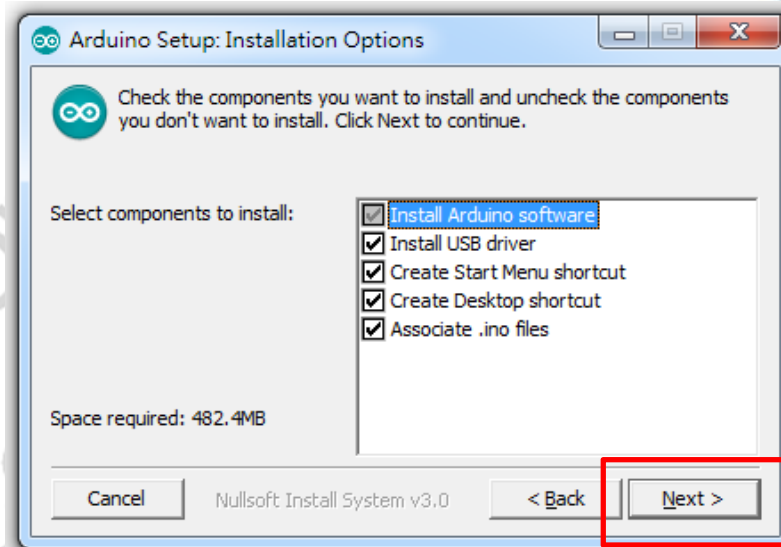
- 打開安裝檔開始安裝

1  arduino-1.8.9-windows.exe

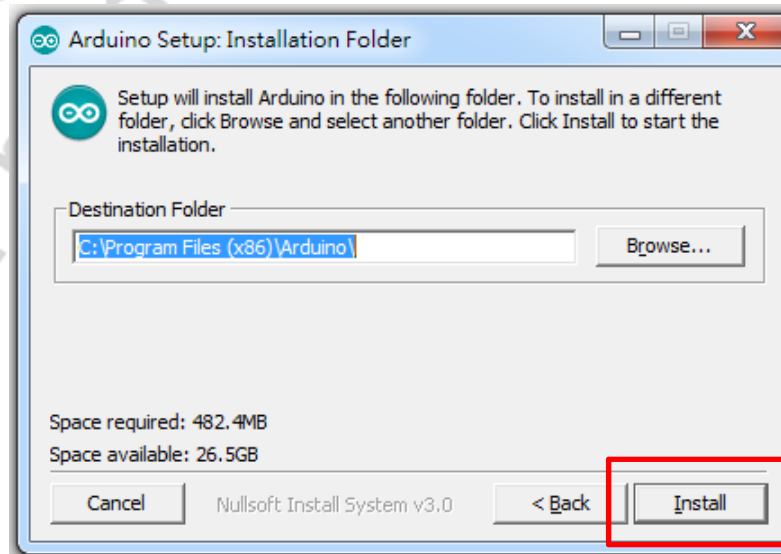
2



3



4



安裝Arduino IDE(6)

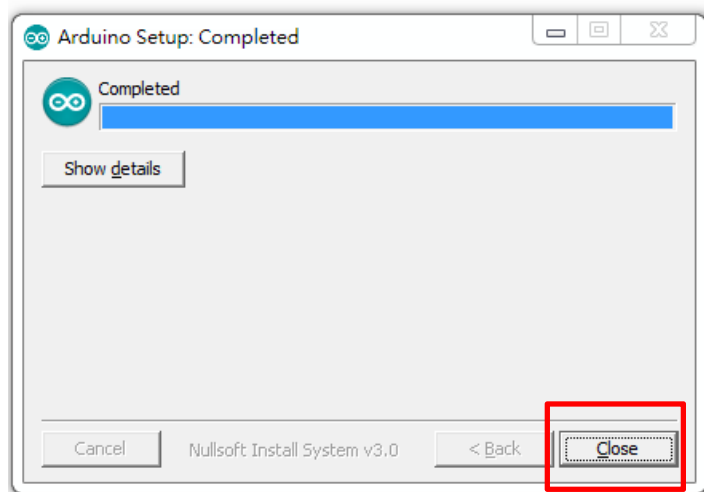
- 打開安裝檔開始安裝

5



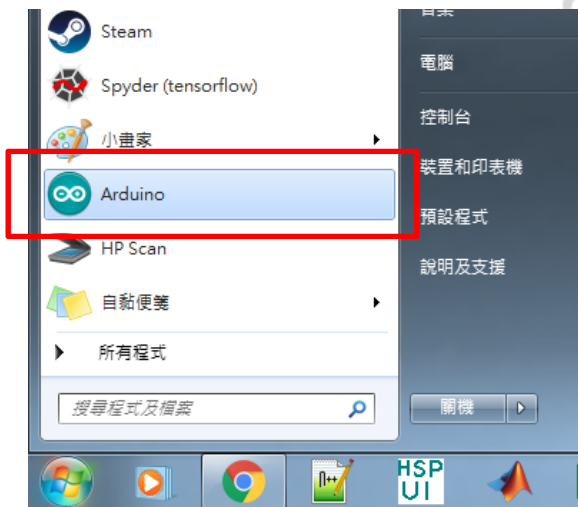
到這一步就把Arduino IDE安裝好了，請大家練習安裝，有問題隨時舉手問助教

6

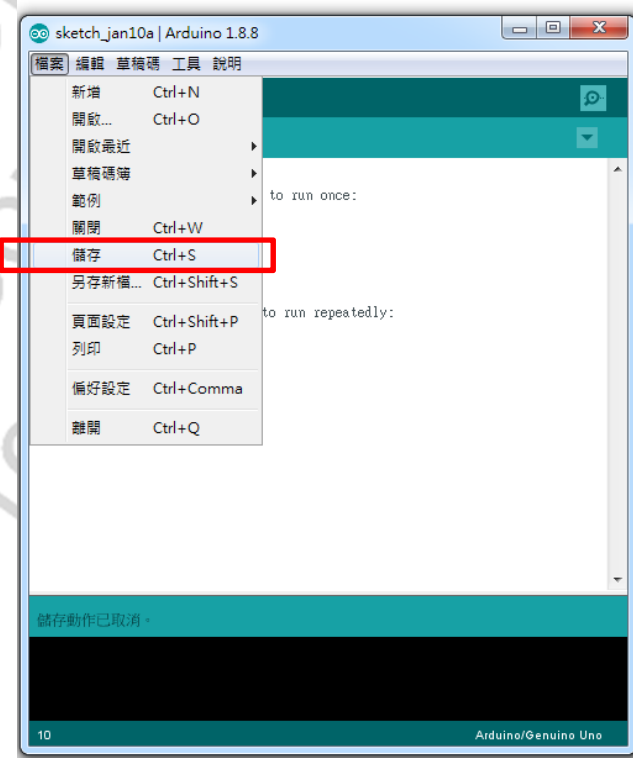


開啟Arduino IDE

- 在桌面、開始選單或者搜尋程式，選取或搜尋“Arduino”



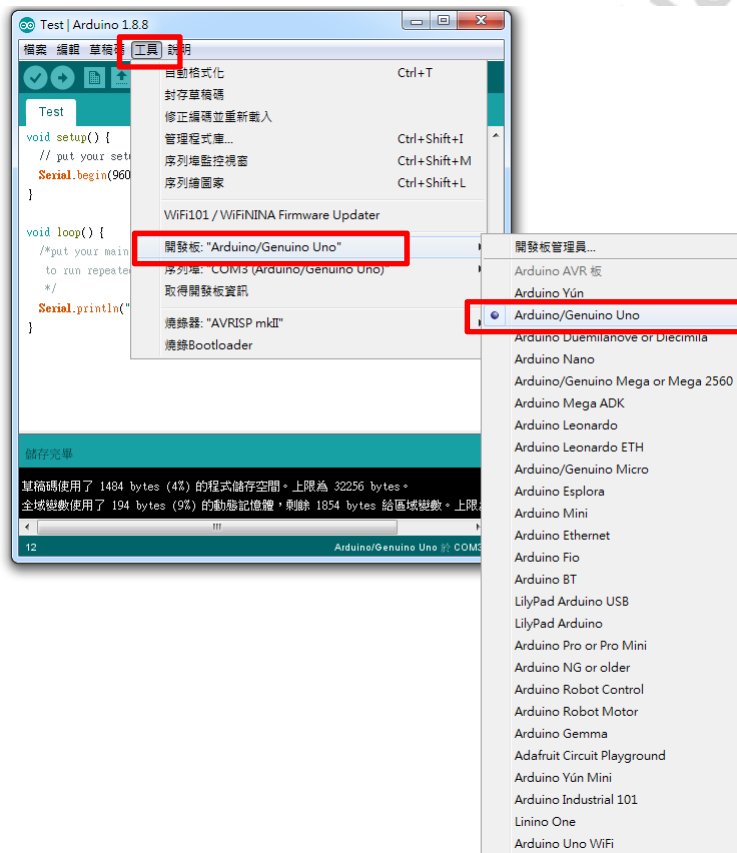
- 開啟後，先儲存檔案
 - 存在桌面，改名稱為 “Test”



確定Arduino連接

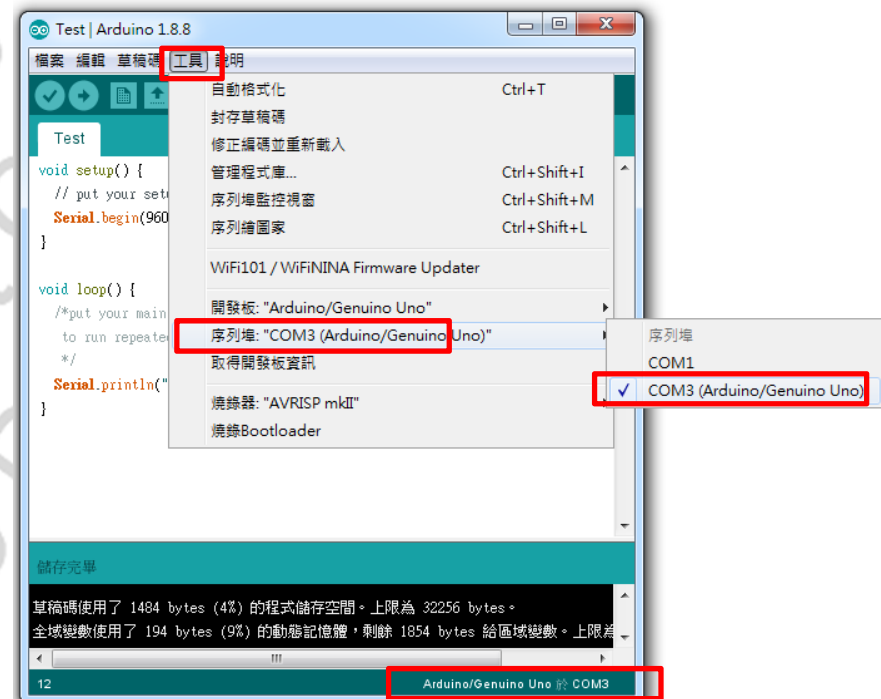
- 將Uno板插入USB，並確認是否連通到正確的板子

□ 工具>開發板>Arduino/Genuino Uno



- 接著，確認序列埠是否選到插Uno的那個USB孔

□ 工具>序列埠
>COM?(Arduino/Genuino Uno)





Arduino的語言

■ Arduino語言架構

- *setup*: 每次程式啟動時，會呼叫這個區域的任務，僅此一次
- *loop*: 執行完*setup*後，會反覆呼叫*loop*區域的任務，一直執行
- 為什麼要這樣子設定呢？想想看嵌入式系統的任務是什麼？

```
檔案 編輯 草稿碼 工具 說明
[Icons] 驗證
Test$
void setup(){ 用大括號包成一個“區域”
    // put your setup code here, to run once:

}

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

}
```


Arduino: Hello World!!

■ *Serial.begin()*

- 設置資料傳輸的速率，單位bits/sec
- 又稱作baud rate (中文：鮑率)
- 和電腦溝通的速率選擇有：
 - 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200

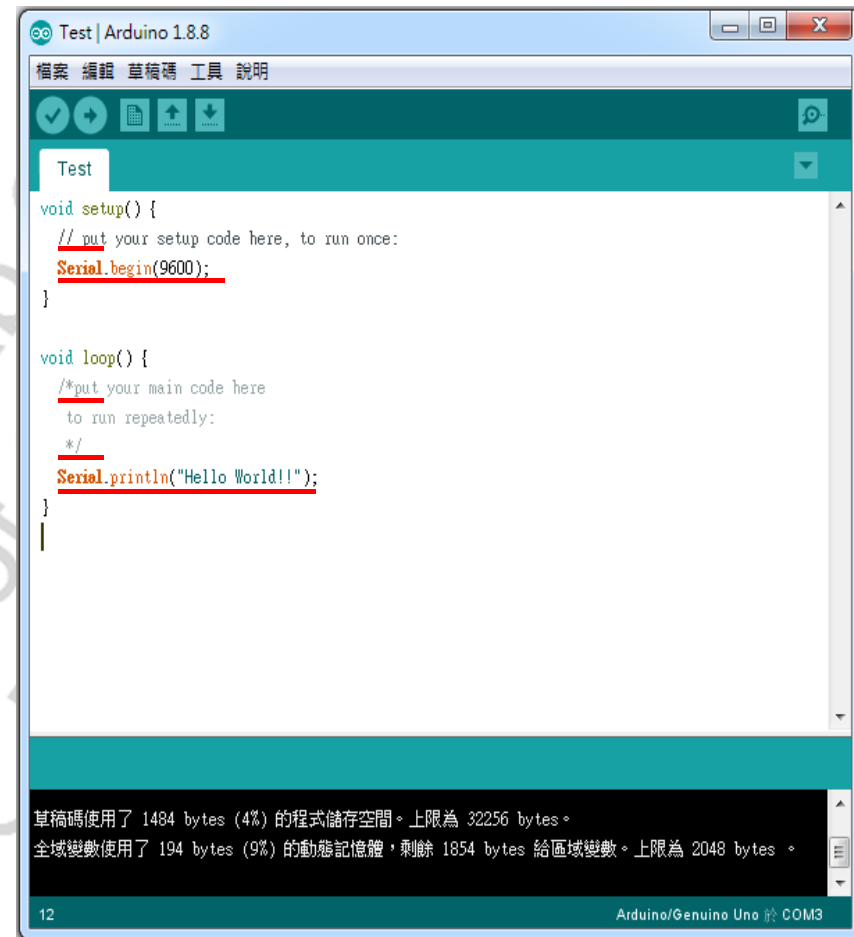
■ *Serial.println()*

- 將括號裡的資訊，透過外部電腦顯示出來
- Serial.print() vs. Serial.println()

■ *//*: 代表單行註解，不會被程式讀到，是給人讀的

■ */* */*: 代表多行註解

■ *;* (分號): 加在每行指令後面



```
Test | Arduino 1.8.8
檔案 編輯 草稿碼 工具 說明

Test

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

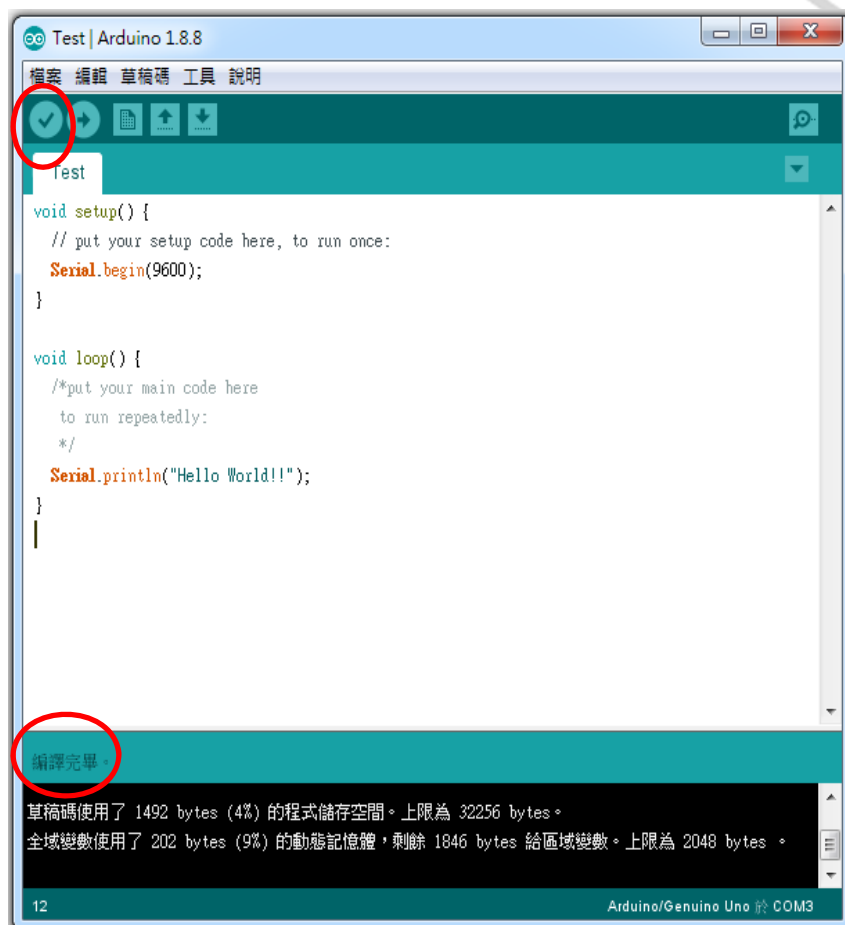
void loop() {
  /*put your main code here
   to run repeatedly:
  */
  Serial.println("Hello World!!");
}
```

草稿碼使用了 1484 bytes (4%) 的程式儲存空間。上限為 32256 bytes。
全域變數使用了 194 bytes (9%) 的動態記憶體，剩餘 1854 bytes 給區域變數。上限為 2048 bytes。

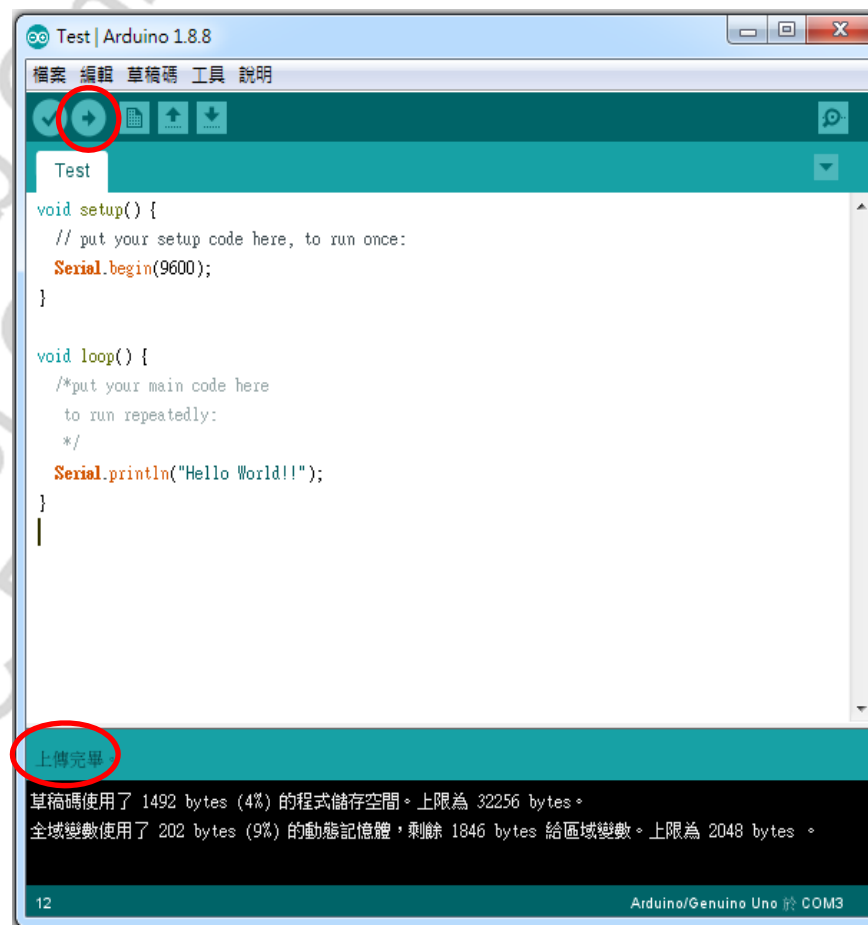
12 Arduino/Genuino Uno COM3

Arduino: Hello World!!

- 按左上角的進行編譯，檢查程式有沒有語法錯誤



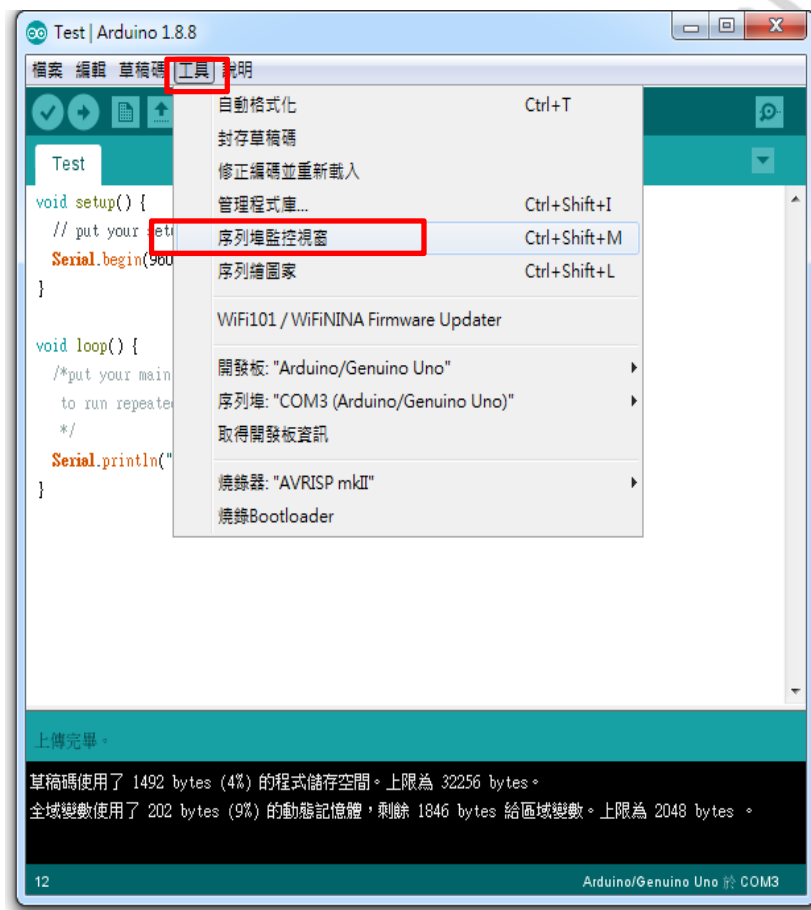
- 接著按旁邊的，將程式碼燒到Uno上



Arduino: Hello World!!

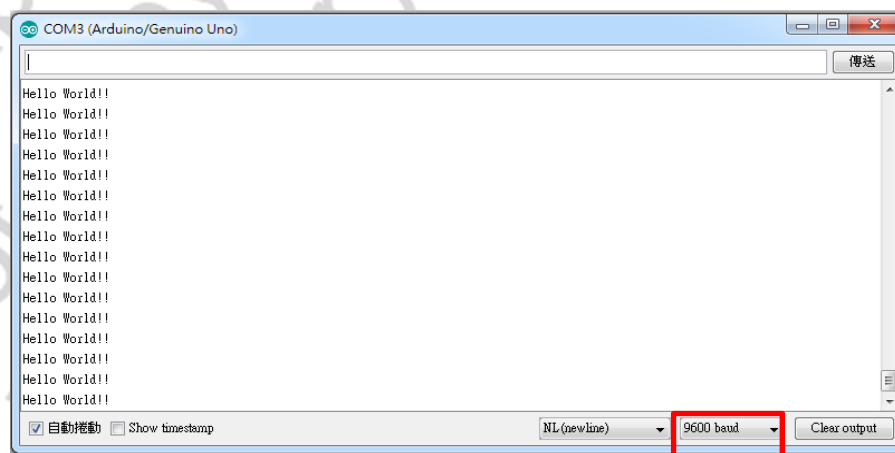
■ 打開序列埠視窗，查看結果

- 工具>序列埠監控視窗



■ 查看結果

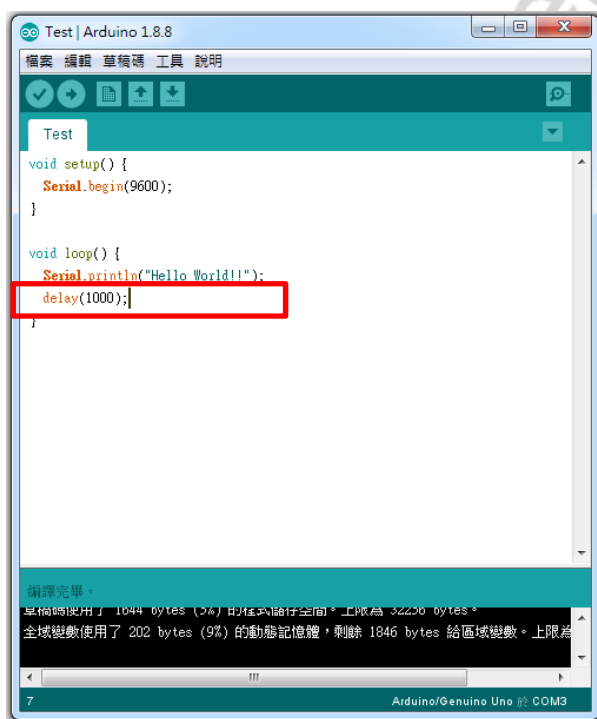
- 可以看到一直不斷的顯示" Hello World!!" 字樣
- 如果把右下角的baud rate更改成其他的速率，會怎麼樣呢？



Arduino: Hello World!!

■ *delay()*

- ❑ 將程式暫停一段時間，括號裡面填入要暫停的時間
- ❑ 括號裡的單位為 ms(毫秒)
- ❑ 1000ms = 1s



```
Test | Arduino 1.8.8
檔案 編輯 草稿碼 工具 說明

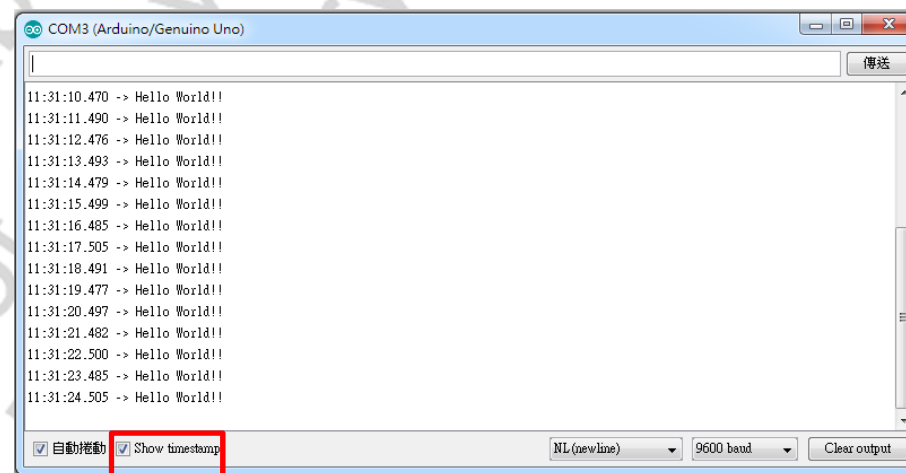
Test
void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.println('Hello World!!');
  delay(1000);
}

編譯完畢。
草稿碼使用了 1044 bytes (5%) 的程式記憶體空間，上限為 32256 bytes。
全域變數使用了 202 bytes (9%) 的動態記憶體，剩餘 1846 bytes 給區域變數。上限為...
```

■ 查看結果，跟剛剛的有什麼不一樣

- ❑ 可以打開Time Stamp功能，看每次的output出來時間是不是你所預期的？



```
COM3 (Arduino/Genuino Uno)
傳送

11:31:10.470 -> Hello World!!
11:31:11.490 -> Hello World!!
11:31:12.476 -> Hello World!!
11:31:13.493 -> Hello World!!
11:31:14.479 -> Hello World!!
11:31:15.499 -> Hello World!!
11:31:16.485 -> Hello World!!
11:31:17.505 -> Hello World!!
11:31:18.491 -> Hello World!!
11:31:19.477 -> Hello World!!
11:31:20.497 -> Hello World!!
11:31:21.482 -> Hello World!!
11:31:22.500 -> Hello World!!
11:31:23.485 -> Hello World!!
11:31:24.505 -> Hello World!!

自動捲動 Show timestamp NL(new line) 9600 baud Clear output
```



Next Step!!

Hello World!!



Hello Arduino!!



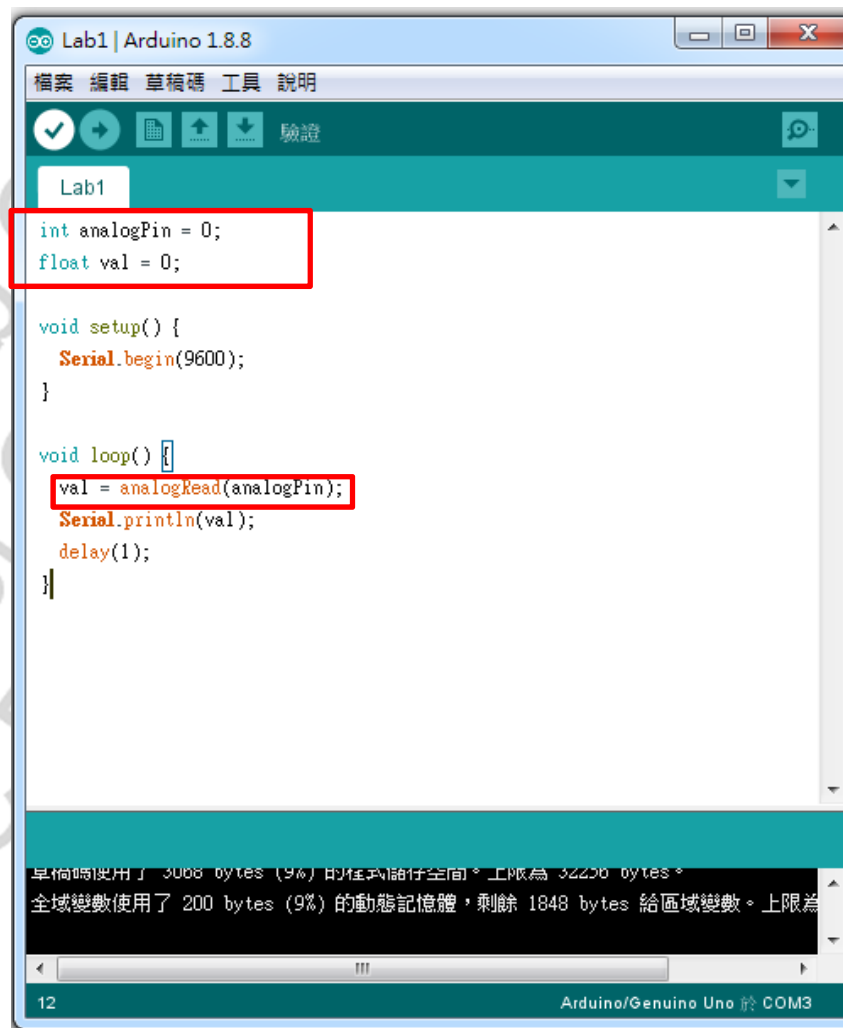


Outline

- 前言
- 設計概念
- 實作
 - Lab1: 讀自己的ECG訊號
 - Lab2: 找到訊號的最大值，並以閾值控制
 - Lab3: 加入時間閾值
 - Lab4: count bpm
- 總結

實作: Lab1

- 運用Trianswer ECG模組來量測自己的心電訊號
- *analogRead()*:
 - ❑ 讀取類比訊號，括號裡填想要讀取的腳位
 - ❑ 類比腳位：0,1,2,3,4,5
- *int, float*:
 - ❑ 宣告變數的型態
 - ❑ int: integer，整數
 - ❑ float，浮點數
 - ❑ 宣告變數的位置有差嗎？
 - global vs. local



```
Lab1 | Arduino 1.8.8
檔案 編輯 草稿碼 工具 說明
[Icons] 驗證
Lab1
int analogPin = 0;
float val = 0;

void setup() {
  Serial.begin(9600);
}

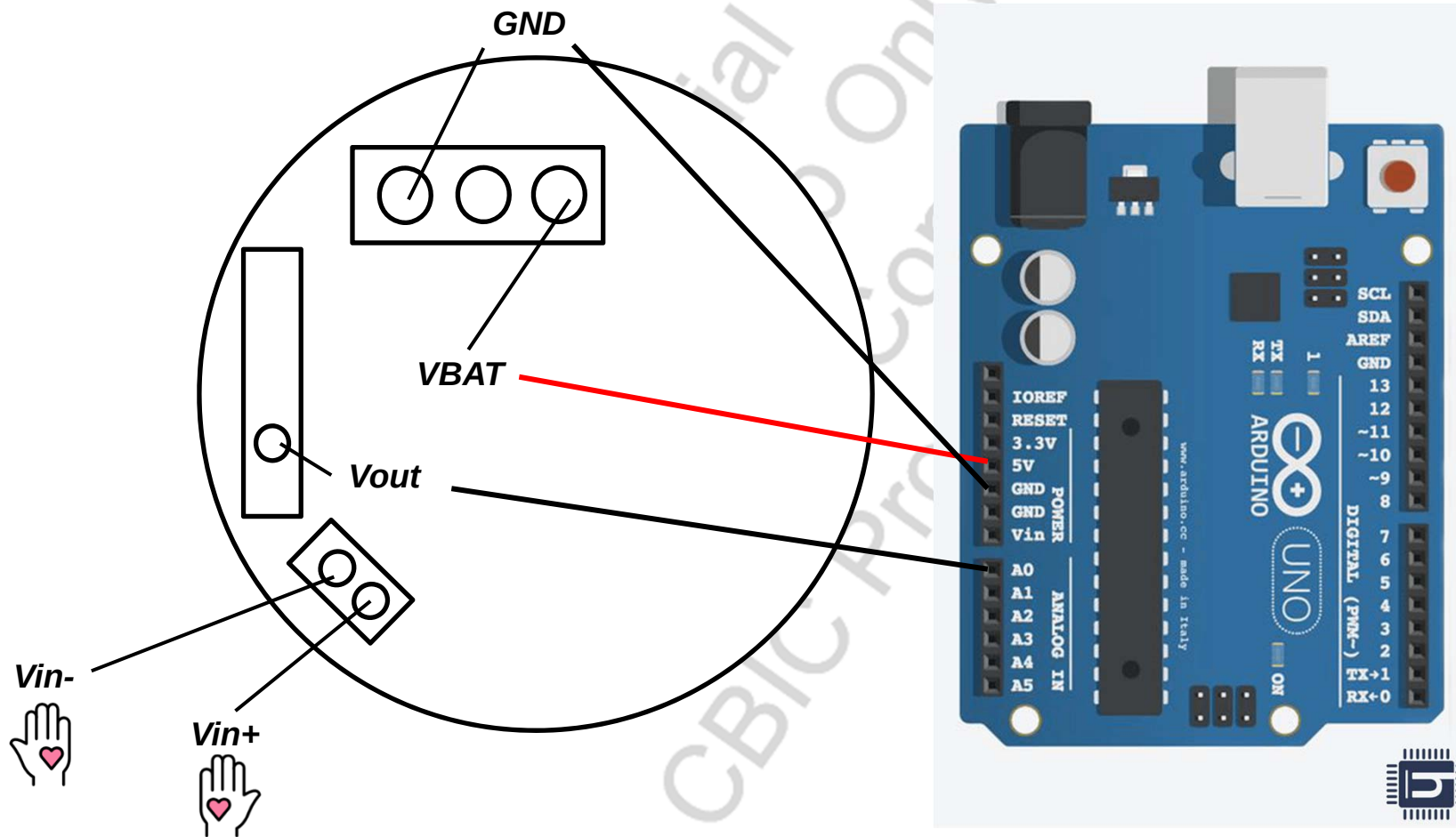
void loop() {
  val = analogRead(analogPin);
  Serial.println(val);
  delay(1);
}
```

草稿碼使用了 3008 bytes (9%) 的程式碼儲存空間。上限為 32256 bytes。
全域變數使用了 200 bytes (9%) 的動態記憶體，剩餘 1848 bytes 給區域變數。上限為

12 Arduino/Genuino Uno 於 COM3

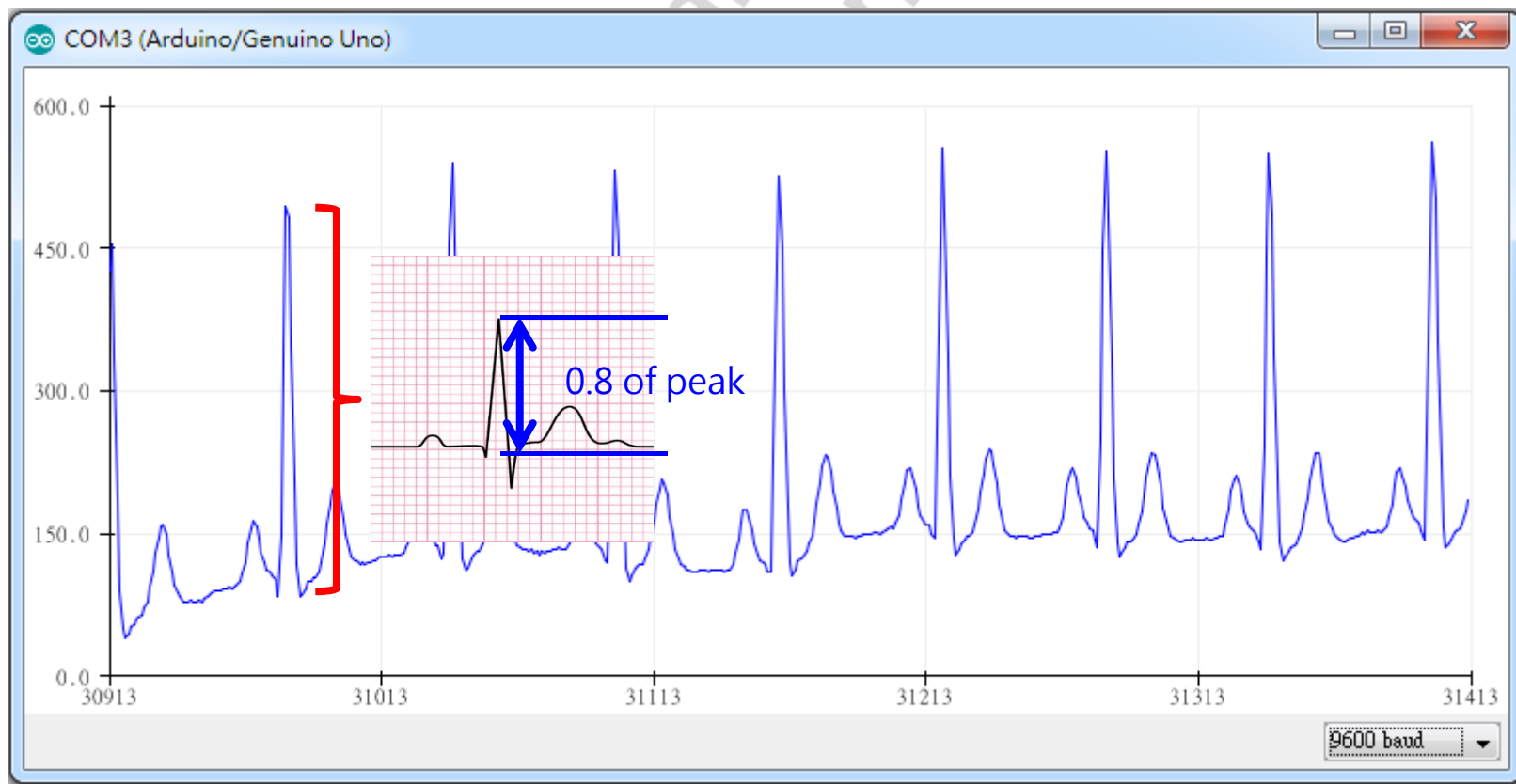
實作: Lab1

- Trianswer ECG模組和 Arduino Uno的接法



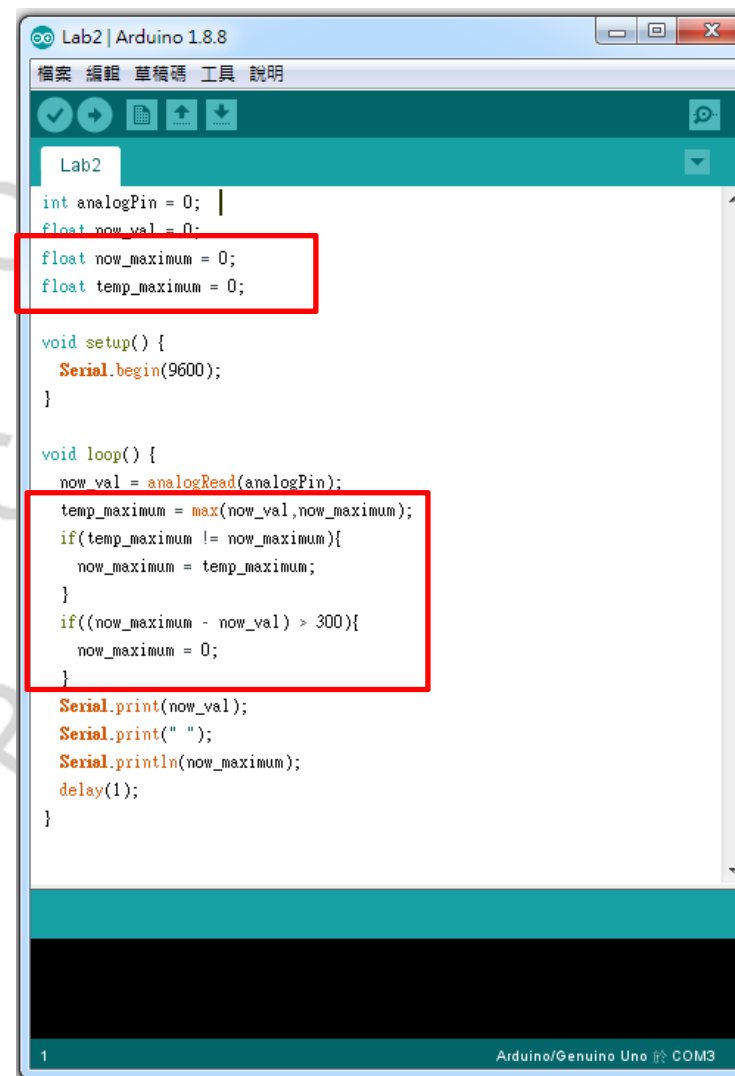
實作: Lab1

- 用序列繪圖家來描繪出自己的心電圖訊號，觀察自己的峰對峰值大約是多少？



實作: Lab2

- 開始找每次的peak
- now_maximum: 現在的最大值
- temp_maximum: 暫時的最大值
- *max()*:
 - $\max(a,b)$, 回傳a和b中較大者
- *if()*:
 - if(條件)
 - { 要執行的任務 }
- $!=, ==, >, <$:
 - 不等於、等於、大於、小於



```
Lab2 | Arduino 1.8.8
檔案 編輯 草稿碼 工具 說明

Lab2

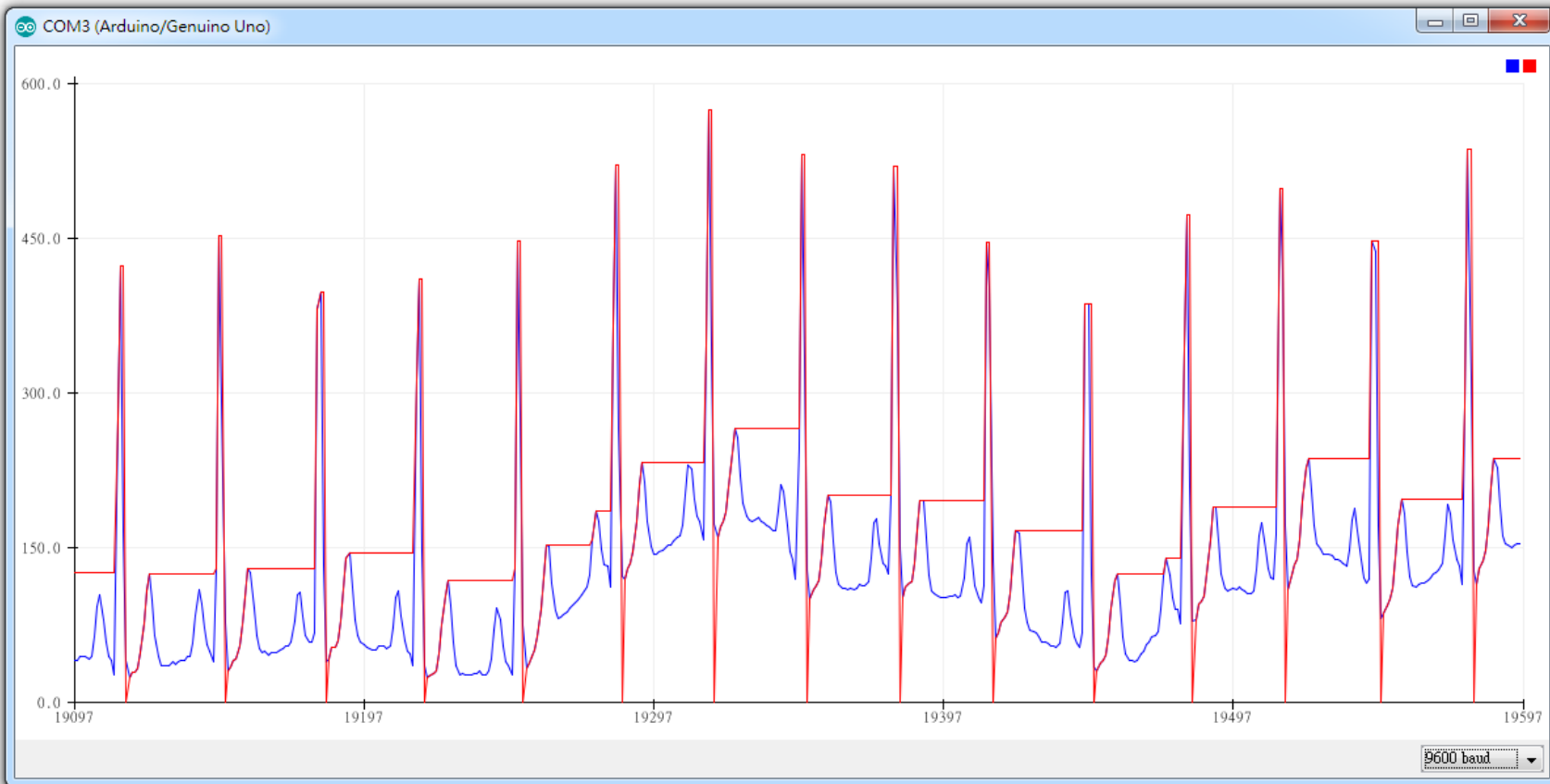
int analogPin = 0;
float now_val = 0;
float now_maximum = 0;
float temp_maximum = 0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  now_val = analogRead(analogPin);
  temp_maximum = max(now_val, now_maximum);
  if(temp_maximum != now_maximum){
    now_maximum = temp_maximum;
  }
  if((now_maximum - now_val) > 300){
    now_maximum = 0;
  }
  Serial.print(now_val);
  Serial.print(" ");
  Serial.println(now_maximum);
  delay(1);
}
```


實作: Lab2

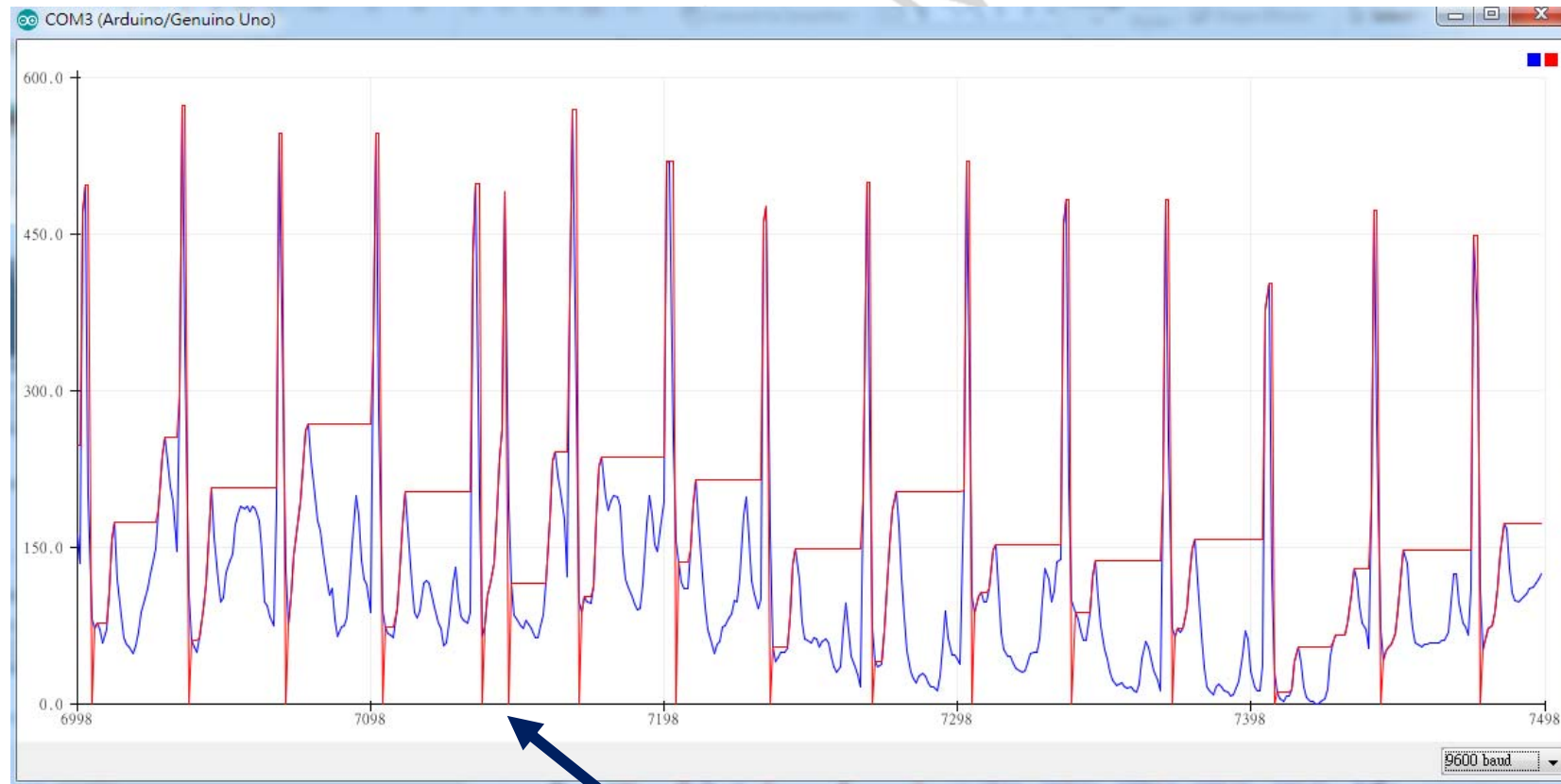
- 用序列繪圖家觀察結果：好像很完美，真的嗎？



實作: Lab2



- 用序列繪圖家觀察結果：有移動雜訊時，有誤判的R peak



因為移動而造成的突波雜訊，誤判成R peak

實作: Lab3

- 現在要加入時間閥值，來避免雜訊或者很高的T波被誤判成R peak
- 前面提到，人的心跳最快不超過200bpm
 - 200bpm
 - => 3.33beats/sec
 - => 0.3sec/beat
 - => 300ms/beat
- 這裡的delay(1)，使得loop區塊的程式跑完一次後，會等1ms，再跑一次loop區塊



```
Lab1 | Arduino 1.8.8
檔案 編輯 草稿碼 工具 說明
Lab1
int analogPin = 0;
float val = 0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  val = analogRead(analogPin);
  Serial.println(val);
  delay(1);
}
```

目前使用了 3008 bytes (9%) 的程式儲存空間。上限為 32256 bytes。
全域變數使用了 200 bytes (9%) 的動態記憶體，剩餘 1848 bytes 給區域變數。上限為

12 Arduino/Genuino Uno 於 COM3

實作: Lab3

- 那麼，我們只要加入counter，每一次loop區塊開始計算現今峰值，再看這次的峰值對到的counter有沒有大於300就好了
- 這樣一來，就能確保每次計算出來的峰值，兩兩相距都有大於300ms了!
- ```
if(條件A){
 if(條件B){
 任務C
 }
}
```

這種結構，稱作“**巢狀結構**”



```
Lab3 | Arduino 1.8.8
檔案 編輯 草稿碼 工具 說明
Lab3
int analogPin = 0;
float now_val = 0;
float now_maximum = 0;
float temp_maximum = 0;
int count = 0;

void setup() {
 Serial.begin(9600);
}

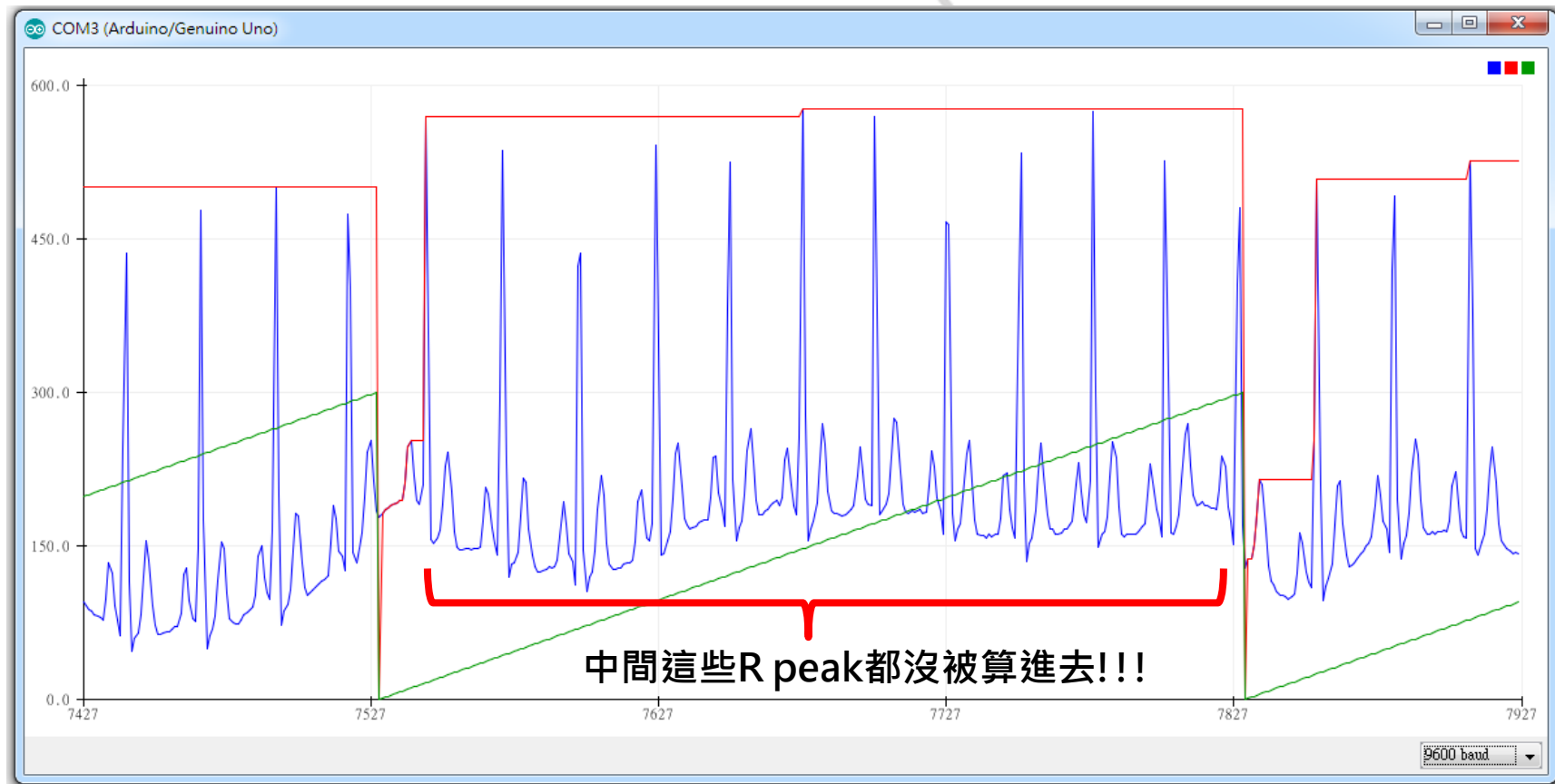
void loop() {
 now_val = analogRead(analogPin);
 count = count + 1;
 temp_maximum = max(now_val, now_maximum);
 if(temp_maximum != now_maximum){
 now_maximum = temp_maximum;
 }
 if((now_maximum - now_val) > 300){
 if(count > 300){
 now_maximum = 0;
 count = 0;
 }
 }
 Serial.print(now_val);
 Serial.print(" ");
 Serial.print(now_maximum);
 Serial.print(" ");
 Serial.println(count);
 delay(1);
}
```

草稿碼使用了 3724 bytes (11%) 的程式碼空間。上限為 32256 bytes。  
全域變數使用了 208 bytes (10%) 的動態記憶體，剩餘 1840 bytes 給區域變數。上限為 18400 bytes。

24 Arduino/Genuino Uno 於 COM3

# 實作: Lab3

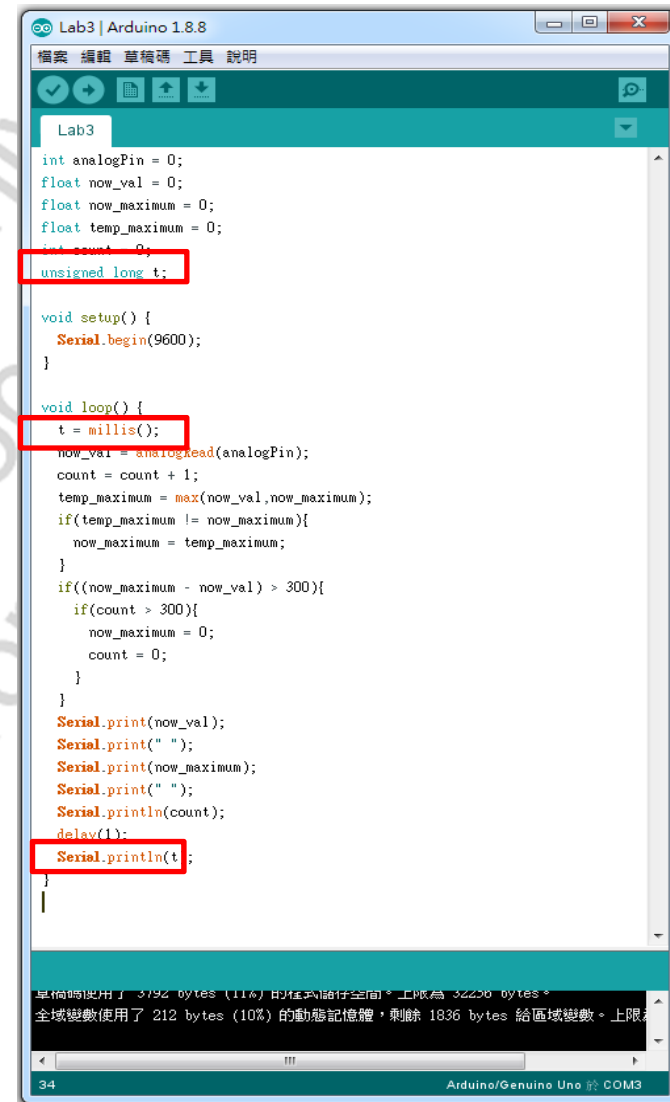
- 用序列繪圖家觀察結果：好像哪裡出錯了....





# 實作: Lab3

- 原因：Arduino Uno、電腦，並不是超級電腦，再怎麼簡單的程式都需要運行時間
- 所以，每次loop相隔時間絕對不是理想的1ms
  - counter如何設定？
- *millis()*:
  - 可以用來看Uno開機到現在所經過的毫秒數
  - 用一個 *unsigned long* 來存這個時間
    - *unsigned*: 非負數
    - *long*: 長整數，可以存比較大的整數



```
Lab3 | Arduino 1.8.8
檔案 編輯 草稿碼 工具 說明

Lab3
int analogPin = 0;
float now_val = 0;
float now_maximum = 0;
float temp_maximum = 0;
int count = 0;
unsigned long t;

void setup() {
 Serial.begin(9600);
}

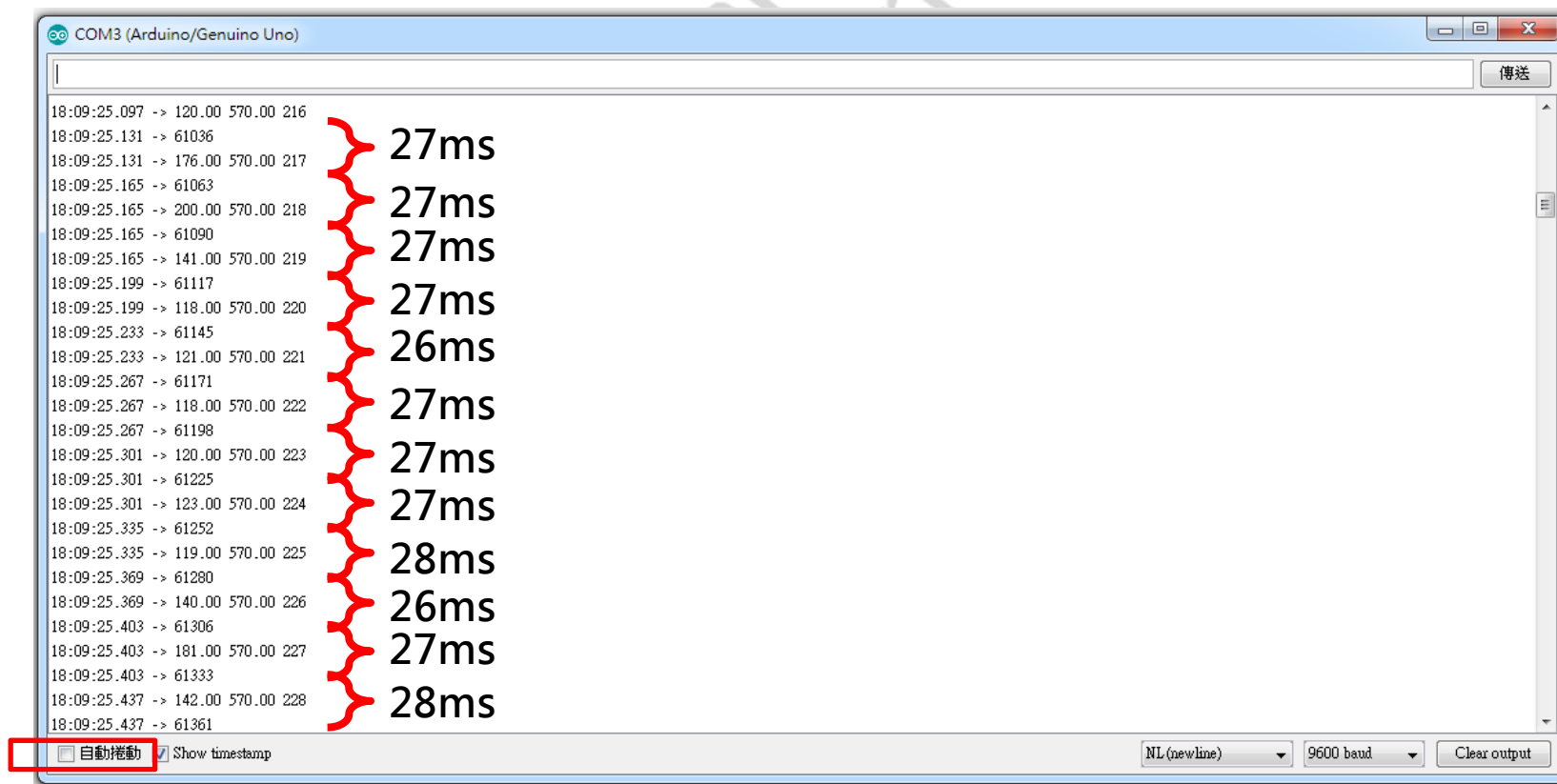
void loop() {
 t = millis();
 now_val = analogRead(analogPin);
 count = count + 1;
 temp_maximum = max(now_val, now_maximum);
 if(temp_maximum != now_maximum){
 now_maximum = temp_maximum;
 }
 if((now_maximum - now_val) > 300){
 if(count > 300){
 now_maximum = 0;
 count = 0;
 }
 }
 Serial.print(now_val);
 Serial.print(" ");
 Serial.print(now_maximum);
 Serial.print(" ");
 Serial.println(count);
 delay(1);
 Serial.println(t);
}
```

草稿碼使用了 3792 bytes (11%) 的程式儲存空間。上限為 32256 bytes。  
全局變數使用了 212 bytes (10%) 的動態記憶體，剩餘 1836 bytes 給區域變數。上限為 18432 bytes。

34 Arduino/Genuino Uno 於 COM3

# 實作: Lab3

- 打開”序列監控視窗”，來看每次loop，相差多久時間，就知道每一次counter相距多久了



# 實作: Lab3

- 平均下來，一次loop所需的時間大約在27ms左右
  - 取25ms，方便計算
- 我們的時間閾值為300ms
  - $300\text{ms}/25\text{ms} = 12$
  - 每次有“疑似”的峰值進來，就確定他是否有大於12個counter
  - 沒有大於的話，就不將其判斷為R peak!

```
Lab3 | Arduino 1.8.8
檔案 編輯 草稿碼 工具 說明

Lab3
int analogPin = 0;
float now_val = 0;
float now_maximum = 0;
float temp_maximum = 0;
int count = 0;
//unsigned long t;

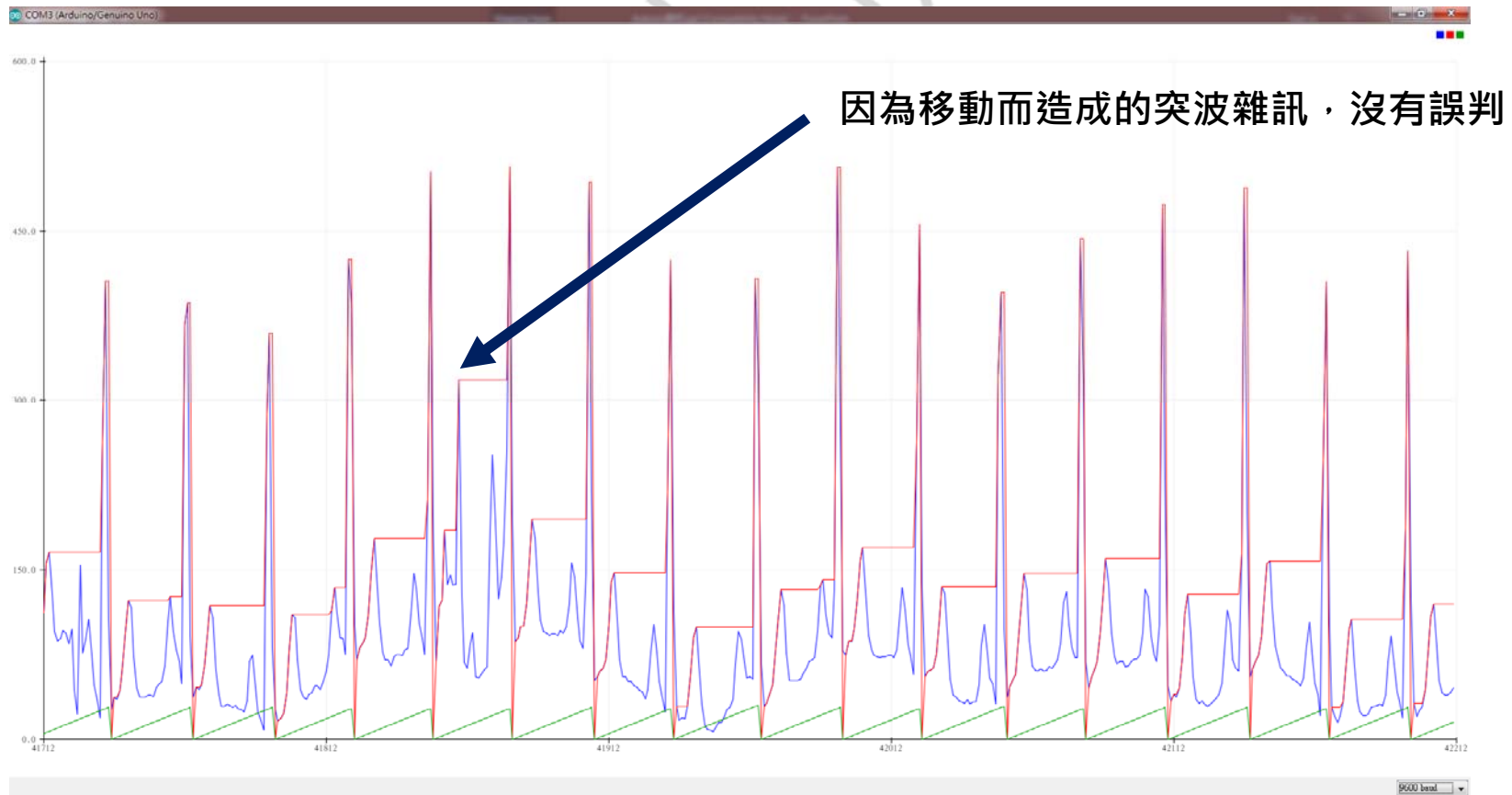
void setup() {
 Serial.begin(9600);
}

void loop() {
 //t = millis();
 now_val = analogRead(analogPin);
 count = count + 1;
 temp_maximum = max(now_val, now_maximum);
 if(temp_maximum != now_maximum){
 now_maximum = temp_maximum;
 }
 if((now_maximum - now_val) > 300){
 if(count > 12){
 now_maximum = 0;
 count = 0;
 }
 }
 Serial.print(now_val);
 Serial.print(" ");
 Serial.print(now_maximum);
 Serial.print(" ");
 Serial.println(count);
 delay(1);
 //Serial.println(t);
}

早編碼使用了 3722 bytes (11%) 的程式儲存空間。上限為 32256 bytes。
全域變數使用了 208 bytes (10%) 的動態記憶體，剩餘 1840 bytes 給區域變數。上限
```

# 實作: Lab3

## ■ 用序列繪圖家觀察結果



# 實作: Lab4

- 算出bpm

- bpm

- $$= (1/(count * count\_time)) * 60$$

- count\_time怎麼決定？還是沿用Lab3中的25ms嗎？

- 註解掉大部分的Serial.print，程式省去了不少時間

- 一個loop大約只花1.15ms
    - count\_time = 0.00115
    - 記得Lab3的時間閾值也要改

```
Lab4 | Arduino 1.8.8
檔案 編輯 草稿碼 工具 說明

Lab4

int analogPin = 0;
float now_val = 0;
float now_maximum = 0;
float temp_maximum = 0;
int bpm = 0;

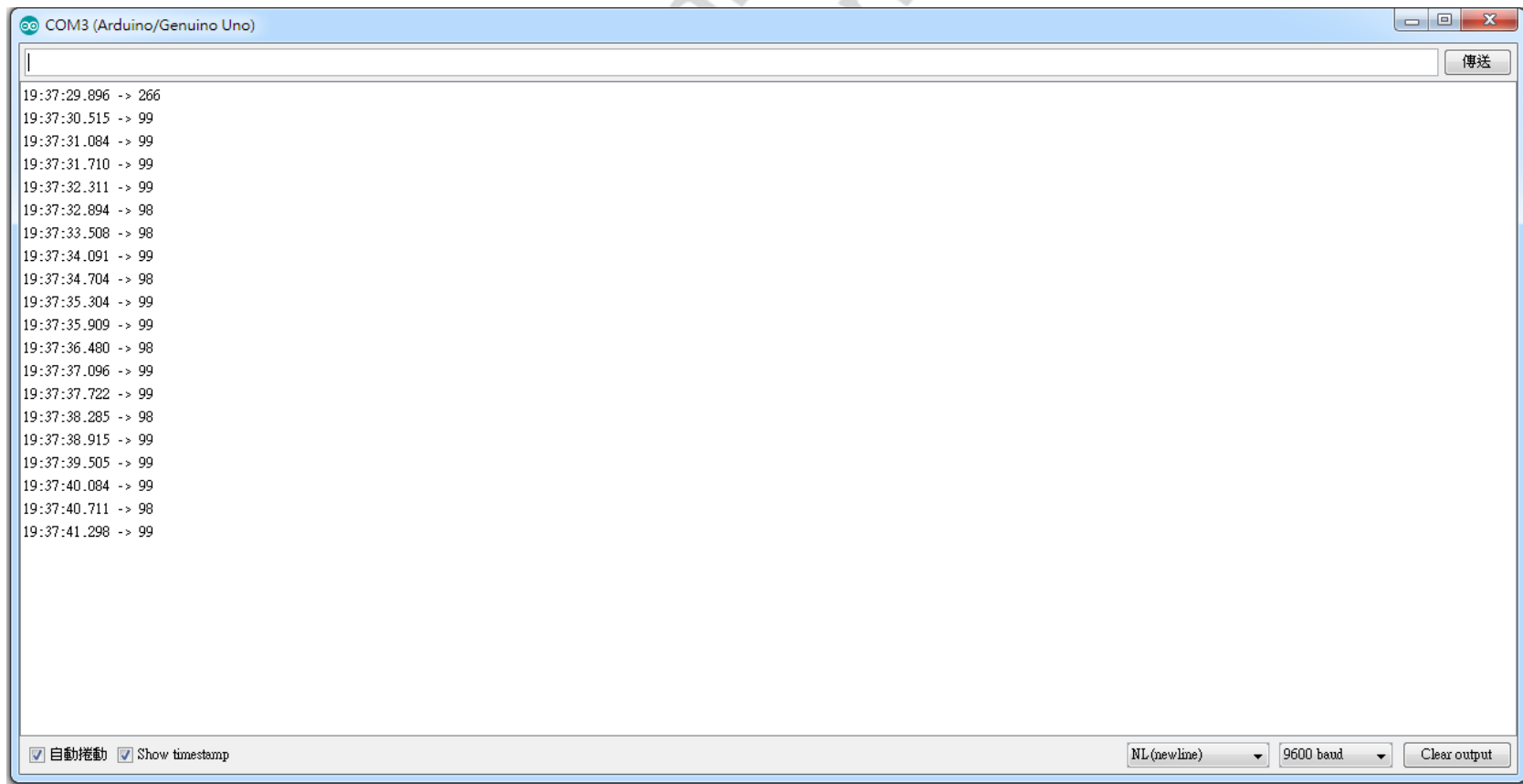
void setup() {
 Serial.begin(9600);
}

void loop() {
 //t = millis();
 now_val = analogRead(analogPin);
 count = count + 1;
 temp_maximum = max(now_val, now_maximum);
 if(temp_maximum != now_maximum){
 now_maximum = temp_maximum;
 }
 if((now_maximum - now_val) > 300){
 if(count > 260){
 now_maximum = 0;
 bpm = (1/(count*0.00115))*60;
 Serial.println(bpm);
 count = 0;
 }
 }
 //Serial.print(now_val);
 //Serial.print(" ");
 //Serial.print(now_maximum);
 //Serial.print(" ");
 //Serial.println(count);
 delay(1);
 //Serial.println(t);
}
```



# 實作: Lab4

- 用序列監控視窗觀察結果：這裡助教用一台心電訊號產生機來模擬，以每分鐘100下來模擬。





## 想想看

- 自己的結果，有沒有和助教的結果不一樣？想想看，把原因記在下面

Confidential  
For CBIC Lab Only  
Do Not Copy  
CBIC Proprietary



# Outline

- 前言
- 設計概念
- 實作
- 總結

Confidential  
For CBIC Lab Only  
Do Not Copy  
CBIC Proprietary



## 總結

- **Arduino**在網路上有很多教學，如果想要更進一步，自學能力是很重要的。
- 嵌入式系統的種類很多，**Arduino**只是其中一種，有興趣的同學也可以去多了解其他種類的嵌入式系統，像是**ARM**。
- 有嵌入式系統的幫助，可以更完善地完成一個系統。
- 這次心率的計算只是個基礎的版本，現階段有很多研究考慮到各種不同的情況。



*I am not the only solution!*



## 參考資料

- 葉難，<http://yehnan.blogspot.com/2012/02/arduino.html>
- Arduino open source，<https://opensource.com/tags/arduino>
- <https://www.arduino.cc/en/Serial/begin>
- <https://www.arduino.cc/en/serial/print>
- <https://www.arduino.cc/reference/en/language/functions/time/delay/>
- <https://hackmd.io/s/Hk7FWc1VB?fbclid=IwAR02XZgkTaGlnSrVorkHeKiuMj0y4a0cR1sk6-8JTV7nzqg8UH6-7bjlNxl>





## Lab1 code

```
int analogPin = 0;
```

```
float val = 0;
```

```
void setup() {
```

```
 Serial.begin(9600);
```

```
}
```

```
void loop() {
```

```
 val = analogRead(analogPin);
```

```
 Serial.println(val);
```

```
 delay(1);
```

```
}
```



## Lab2 code

```
int analogPin = 0;
float now_val = 0;
float now_maximum = 0;
float temp_maximum = 0;
```

```
void setup() {
 Serial.begin(9600);
}
```

```
void loop() {
 now_val = analogRead(analogPin);
 temp_maximum =
 max(now_val, now_maximum);
 if(temp_maximum != now_maximum){
 now_maximum = temp_maximum;
 }
 if((now_maximum - now_val) > 300){
 now_maximum = 0;
 }
 Serial.print(now_val);
 Serial.print(" ");
 Serial.println(now_maximum);
 delay(1);
}
```



# Lab3 code

```
int analogPin = 0;
float now_val = 0;
float now_maximum = 0;
float temp_maximum = 0;
int count = 0;
//unsigned long t;

void setup() {
 Serial.begin(9600);
}

void loop() {
 //t = millis();
 now_val = analogRead(analogPin);
 count = count + 1;
 temp_maximum = max(now_val, now_maximum);
 if(temp_maximum != now_maximum){
 now_maximum = temp_maximum;
 }
 if((now_maximum - now_val) > 300){
 if(count > 12){
 now_maximum = 0;
 count = 0;
 }
 }
 Serial.print(now_val);
 Serial.print(" ");
 Serial.print(now_maximum);
 Serial.print(" ");
 Serial.println(count);
 delay(1);
 //Serial.println(t);
}
```



# Lab4 code

```
int analogPin = 0;
float now_val = 0;
float now_maximum = 0;
float temp_maximum = 0;
int count = 0;
int bpm = 0;
unsigned long t;

void setup() {
 Serial.begin(9600);
}

void loop() {
 //t = millis();
 now_val = analogRead(analogPin);
 count = count + 1;
 temp_maximum = max(now_val, now_maximum);
 if(temp_maximum != now_maximum){
 now_maximum = temp_maximum;
 }
 if((now_maximum - now_val) > 300){
 if(count > 260){
 now_maximum = 0;
 bpm = (1/(count*0.00115))*60;
 Serial.println(bpm);
 count = 0;
 }
 }
 //Serial.print(now_val);
 //Serial.print(" ");
 //Serial.print(now_maximum);
 //Serial.print(" ");
 //Serial.println(count);
 delay(1);
 //Serial.println(t);
}
```



# 預錄訊號以及存取訊號的code

```
■ uint8_t data[] =
{131,130,129,130,131,130,131,132,131,131,131,131,131,132,131,131,133,130,132,132,131,130,130,132,131,130
,131,132,132,131,132,131,129,130,130,132,131,130,129,130,130,130,128,130,130,129,129,130,129,128,128,128
,128,128,128,127,129,127,128,126,127,127,128,128,130,129,131,129,128,129,129,129,131,131,131,130,133,132
,134,135,135,136,134,135,135,132,132,132,133,130,131,128,128,127,124,127,124,124,124,124,123,123,124,124
,123,123,123,123,123,122,125,122,124,123,123,124,122,122,124,122,122,124,122,122,126,128,134,142,150,163
,174,190,202,211,216,206,190,162,136,106,83,69,60,56,57,58,62,65,71,73,75,86,93,100,108,111,114,117,119,11
9,121,123,123,126,126,127,126,128,128,129,130,130,130,131,131,132,132,133,132,133,134,134,134,134,134,13
5,136,135,137,136,137,136,135,137,137,138,138,137,137,139,139,138,139,140,138,140,141,139,140,142,142,14
2,142,141,143,142,143,144,143,143,146,146,145,147,146,147,147,148,148,148,147,149,148,149,148,150,148,15
0,150,149,150,148,150,149,149,148,146,148,147,144,146,145,143,143,141,139,139,138,136,137,135,136,133,13
4,132,130,130,128,130,130,127,129,128,128,126,127,127,126,127,126,126,125,127,124,126,126,124,124,125,12
4,125,125,125,125,124,125,125,124,125,125,126,125,126,125,125,125,126,125,126,125,126,128,125,127,125,12
5,129,125,127,126,128,127,127,127,126,128,127,126,127,126,126,127,127,126,126,126,126,125,126,126,126,12
6,126,124,124,124,125,124,124,123,124,124,125,125,124,126,126,125,125,126,125,125,127,129,129,130,130,13
1,132,132,134,132,132,132,131,130,131,128,128,128,127,125,124,124,124,122,120,120,121,120,120,120,119,12
0,119,121,121,118,119,119,119,120,119,119,117,121,118,119,118,119,119,120,117,118,122,126,130,140,146,16
0,173,186,197,208,212,199,178,151,124,97,74,61,54,52,52,53,60,62,66,68,76,85,90,98,104,108,111,114,114,118
,118,118,118,121,121,123,124,124,124,124,126,126,127,126,129,128,127,129,129,130,129,129,130,130,131,131
,132,131,132,133,132,132,133,135,133,135,133,136,135,136,136,136,136,138,136,137,137,139,139,139,138,139
,139,140,140,140,142,142,141,143,143,142,143,143,146,146,146,145,144,146,145,147,146,148,148,146,148,148
,148,149,148,148,147,147,148,147,144,146,145,141,141,141,141,139,137,137,137,135,135,133,132,131,130,132
,129,128,128,128,126,126,126,124,124,125,125,124,123,124,125,125,125,124,125,126,127,126,125,126,125,124
,126,125,126,126,127,127,127,128,129,129,128,131,130,132,130,130,130,130,129,128
};
```





# 預錄訊號以及存取訊號的code

- 如果要加入雜訊 複製下面這段到剛剛的那一長串資料裡  
77,84,122,137,158,167,199,170,144,130,111,96,72,
- ```
//Arduino coding
uint8_t data[] = {...}
int i = 0;
int analogPin = 0;
float val = 0;
int now_maximum = 0;
int temp_maximum = 0;
int count = 0;
int bpm = 0;

void setup(){
  Serial.begin(9600);
}
```
- ```
void loop(){
 if(i == 600){
 i = 0;
 }
 val = data[i];
 i = i + 1;
 count = count + 1;
 temp_maximum = max(val,now_maximum);
 if(temp_maximum != now_maximum){
 now_maximum = temp_maximum;
 }
 if((now_maximum - val) > 120){
 if(count > 250){
 bpm = (1/(count*0.002))*60;
 Serial.println(bpm);
 now_maximum = 0;
 count = 0;
 }
 }
 //Serial.print(val);
 //Serial.print(" ");
 //Serial.print(now_maximum);
 //Serial.print(" ");
 //Serial.println(count);
 delay(1);
}
```



Thank Your Listening

# 生醫穿戴式裝置開發課程模組

## 課程: APP實作

---

Presenter : 助教 Hsin

Date : 2020/01/30~2021/12/31





# 大綱

- Lab 1 : App Inventor 2 環境介紹、手機模擬
- Lab 2 : 元件與函式庫、藍芽連接功能
- Lab 3 : 應用心律演算法
- Lab 4 : 加入波形圖

# APP實作

## Lab 1 : App Inventor 2 環境介紹 、手機模擬

---

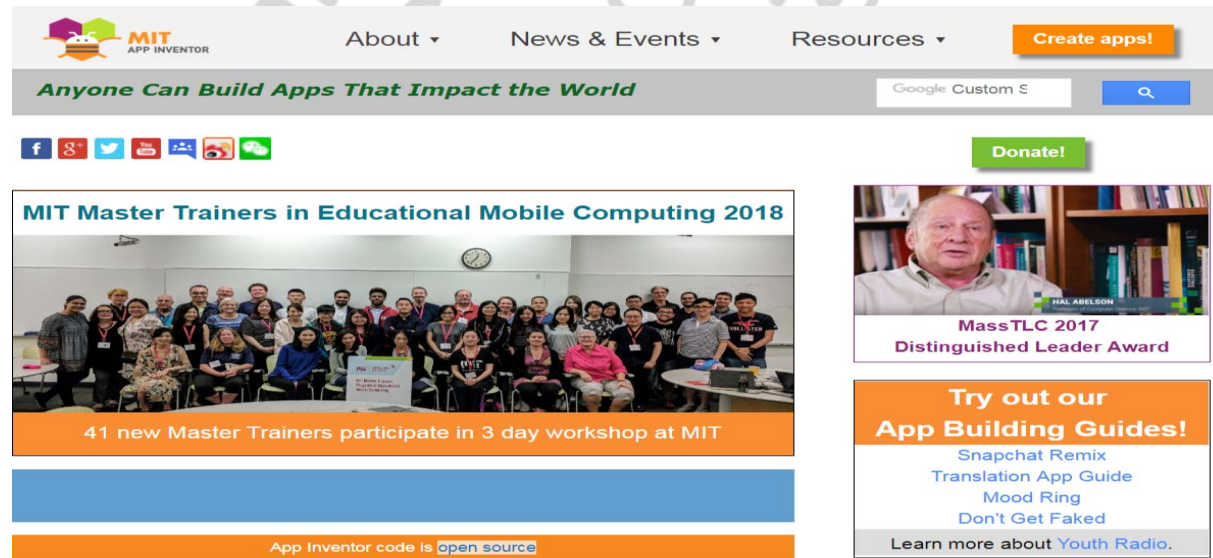




# Lab 1-1

## ■ App Inventor 2 簡介

- ❑ 原Google開發，後MIT接手
- ❑ MIT open source
- ❑ 利用Chrome開發App的雲端環境



# Lab 1-1

- 實驗步驟：1. 網頁搜尋 app inventor 2 → 登入google 帳戶 → 登入成功
- 2. 選擇Start new project→ 專案名稱輸入Trianswer\_座號  
(Ex : Trianswer\_000)

Google Accounts

The application MIT AppInventor Version 2 is requesting permission to access your Google Account.

Please select an account that you would like to use.

● [redacted]@gmail.com

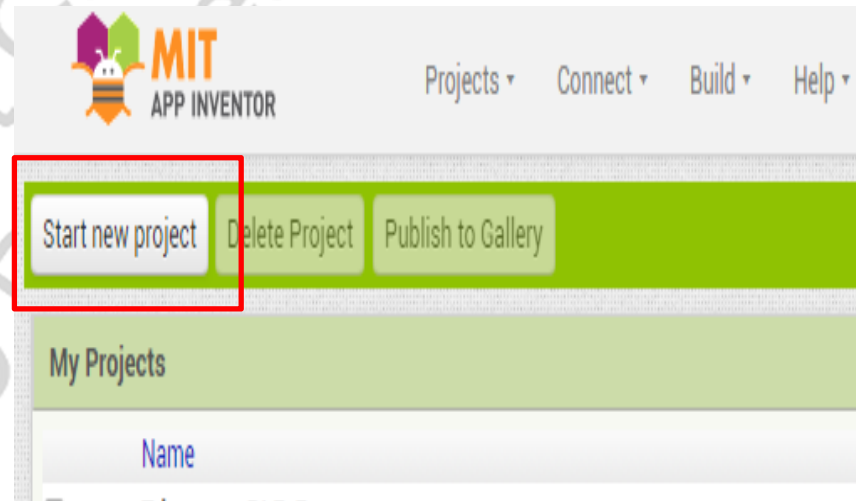
Google is not affiliated with the contents of MIT AppInventor Version 2 or its owners. If you sign in, Google will address with MIT AppInventor Version 2 but not your password or any other personal information.

Allow

No thanks

[Sign in to another account](#)

☒ Remember this approval for the next 30 days



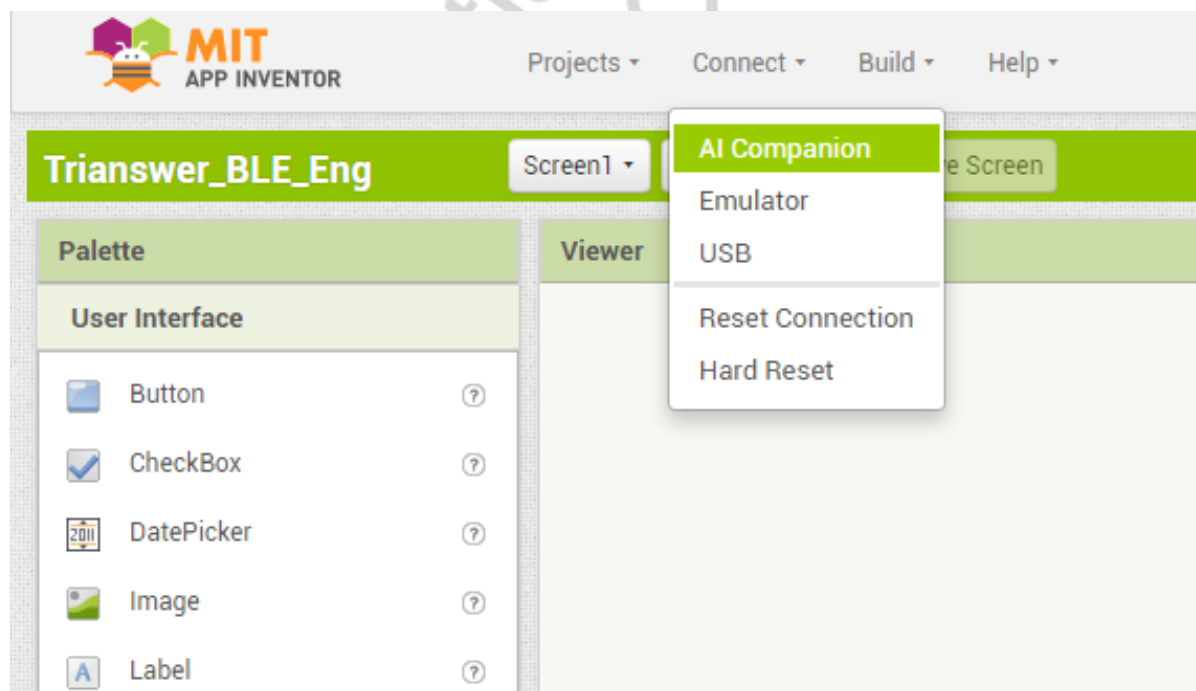
# Lab 1-1

- 實驗步驟：3. 到Play商店下載 MIT AI2 Companion



# Lab 1-1

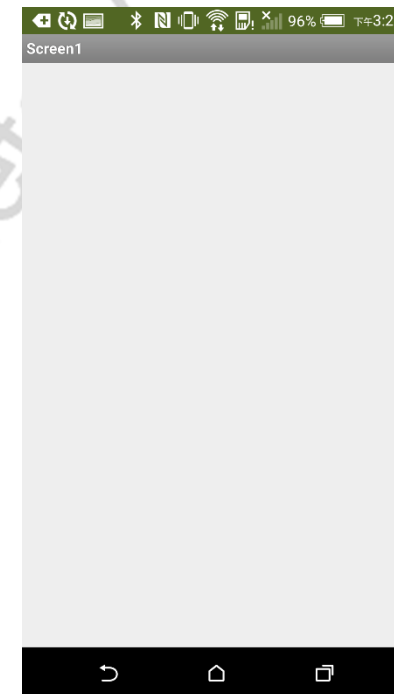
- 實驗步驟：4. 回到專案，點選Connect → AI Companion





# Lab 1-1

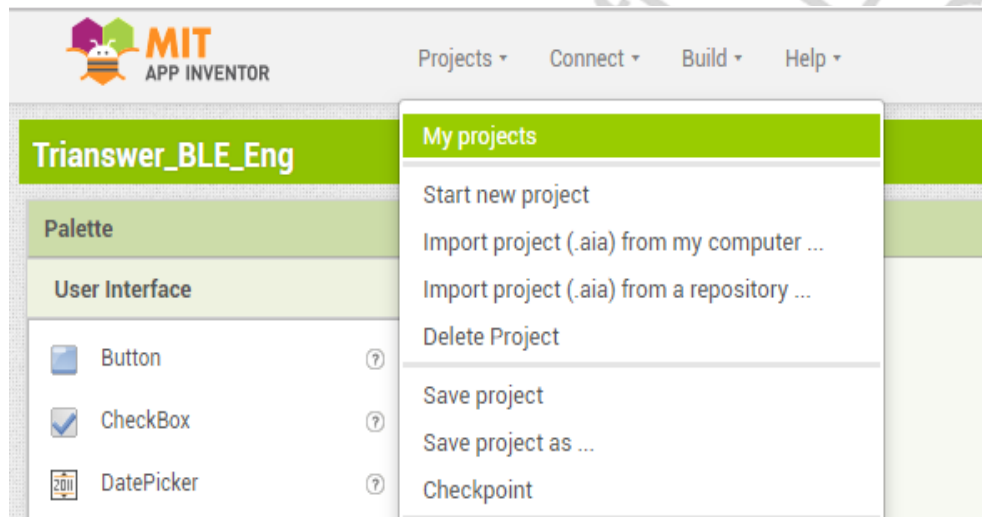
- 實驗步驟：5. 開啟手機App (MIT AI2 Companion)，掃描 QR Code，即可模擬成功。(目前為空白頁面)





# Lab 1-2

- 實驗目的：學習如何開啟舊檔。
- 實驗步驟：點選Projects → Import project(.aia) from my computer → 選擇要上傳的專案 → 匯入成功



# APP實作

## Lab 2 : 元件與函式庫 、藍芽連接功能

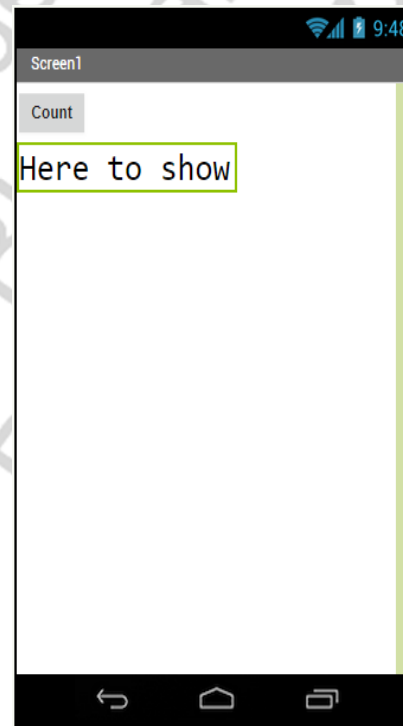
---





## Lab 2-1

- **實驗目的：**實作一個簡易的計數器，新增一個按鈕以及一個標籤，加入其對應的函式，每點按一次按鈕，標籤上的數字就往上加一，將結果顯示在標籤上。



# Lab 2-1

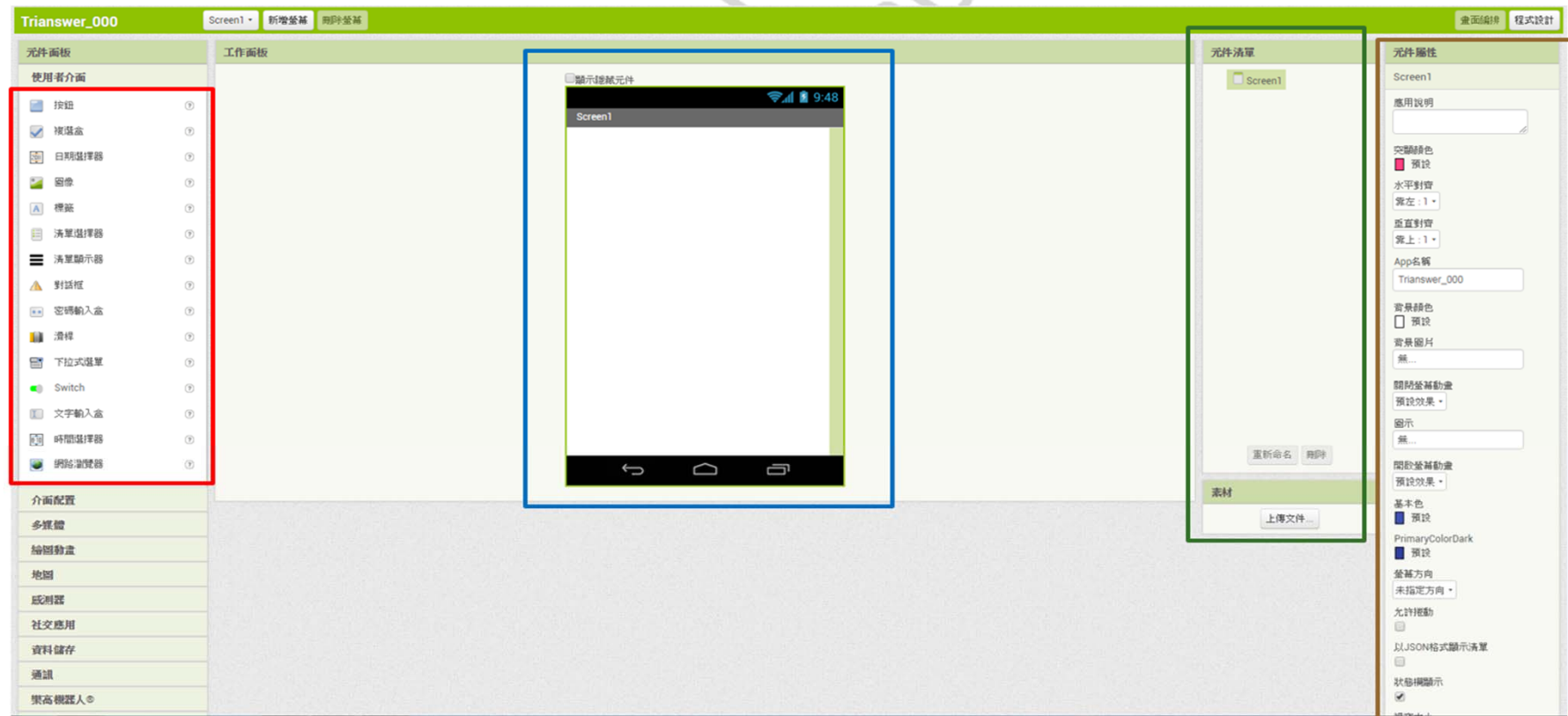
- 實驗步驟：1. 進入專案畫面 → 嘗試新增元件

各類元件

APP螢幕顯示

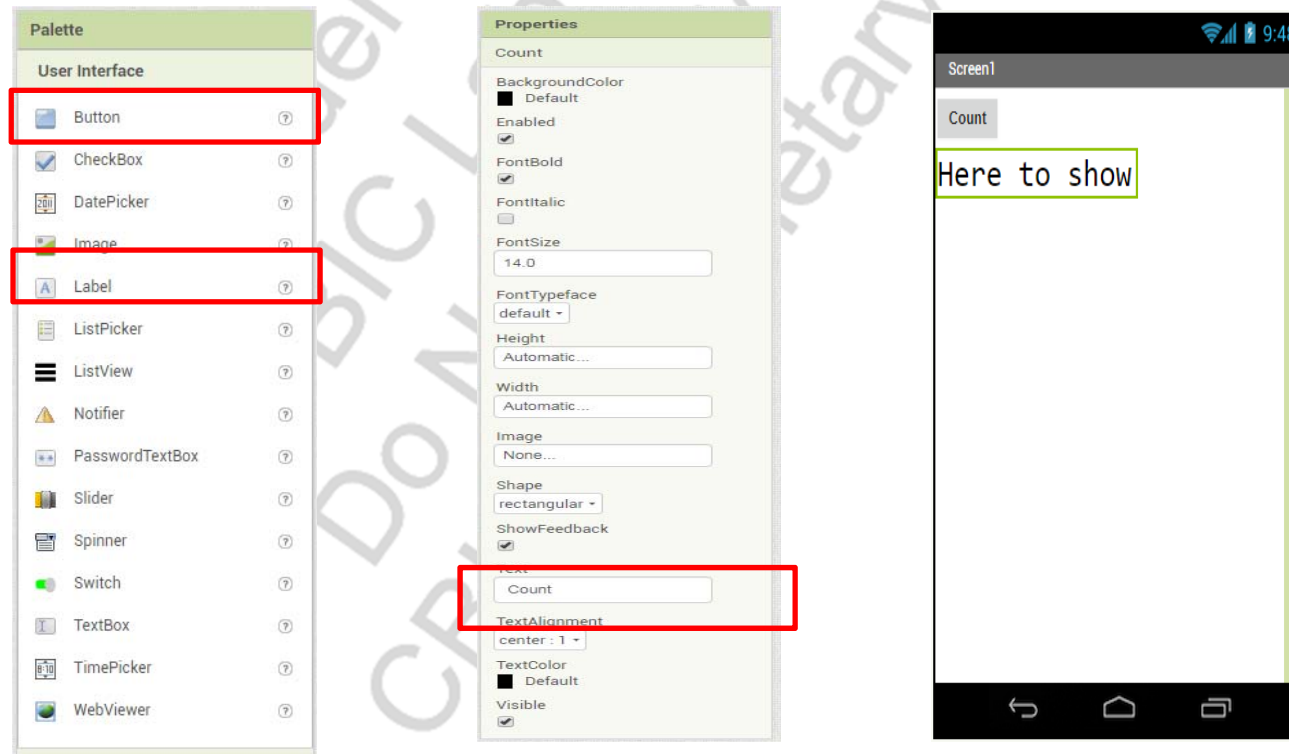
各元件相  
互關係

選取元件  
屬性



## Lab 2-1

- 實驗步驟：2. 新增按鈕方法：選取**按鈕鍵**，**按住滑鼠右鍵拖曳**至App螢幕顯示區的水平配置區塊內，在右側元件屬性區更改按鈕名稱。新增標籤也是相同方法。







## Lab 2-1

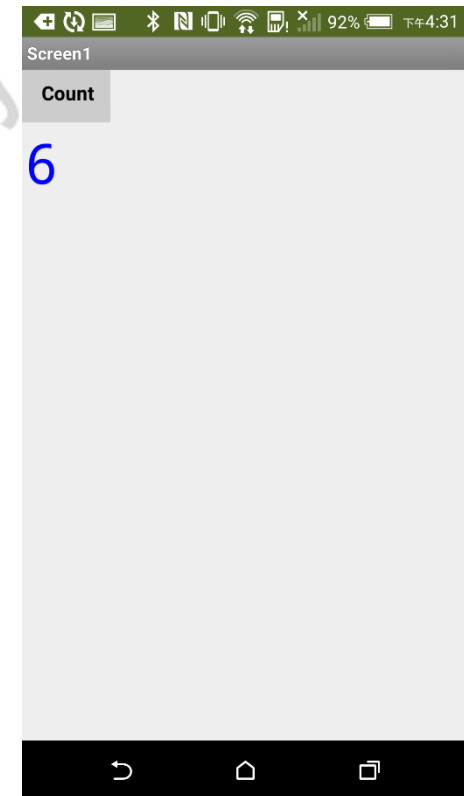
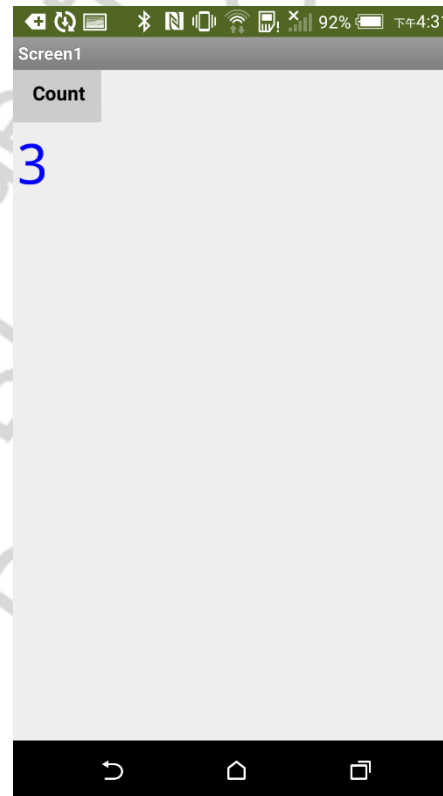
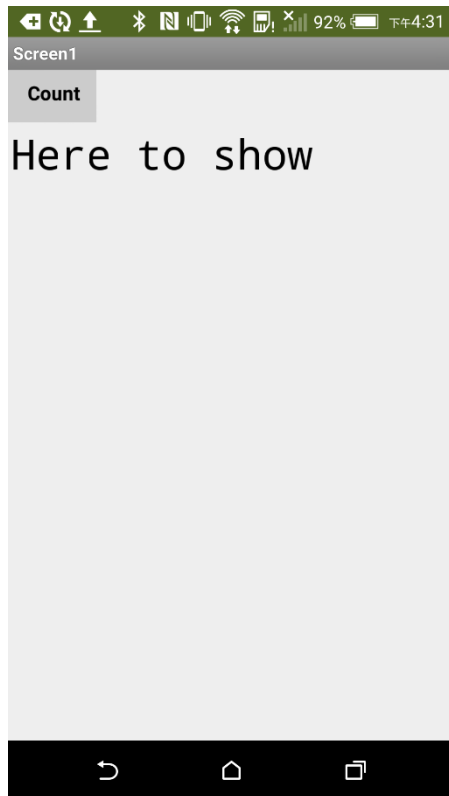
- 實驗步驟：3. 進入程式設計頁面，選擇對應的函式，完成實作。





## Lab 2-1

- 實驗步驟：4. 連接手機進行模擬，確認是否能正常執行





## 附件 A – 元件Button

### ■ 屬性

- BackgroundColor、Enable、Text、Visible.....

### ■ 事件

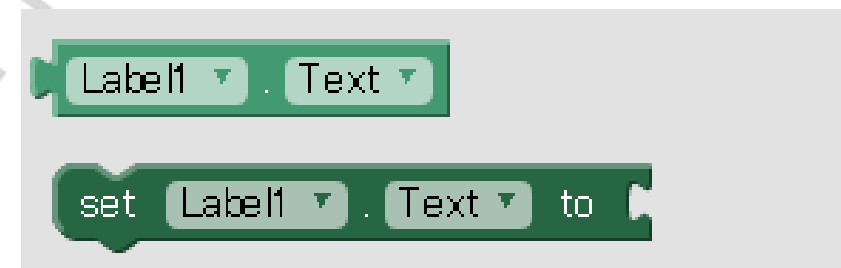
- Click()、GotFocus()、LostFocus().....





## 附件A – 元件Label

- 屬性
  - BackgroundColor、Text、Visible.....

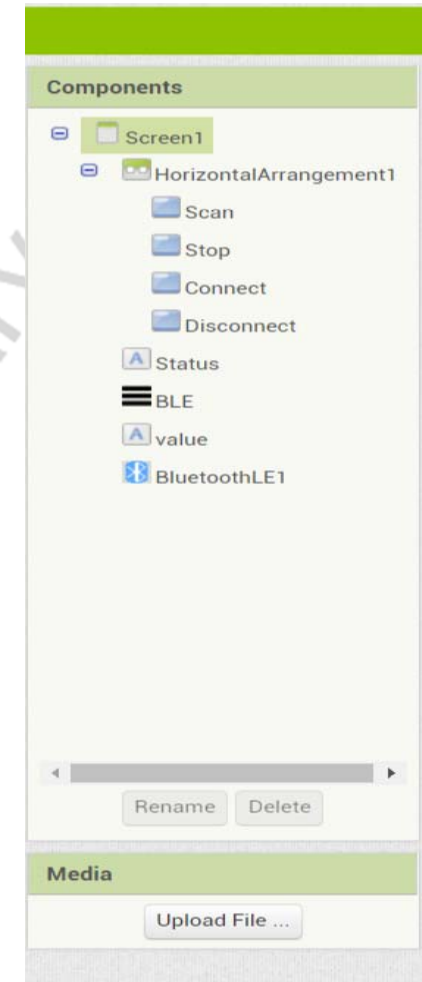




# 附件A – 元件Image

## ■ 屬性

- Picture、Visible、Width.....





## 附件A – 元件ListView

- 屬性

- ElementsFromString、Enable、Visible.....

- 事件

- AfterPicking()

```
set ListView1 . ElementsFromString to
```

```
when ListView1 . AfterPicking
do
```





## 附件A – 元件Web

### ■ 屬性

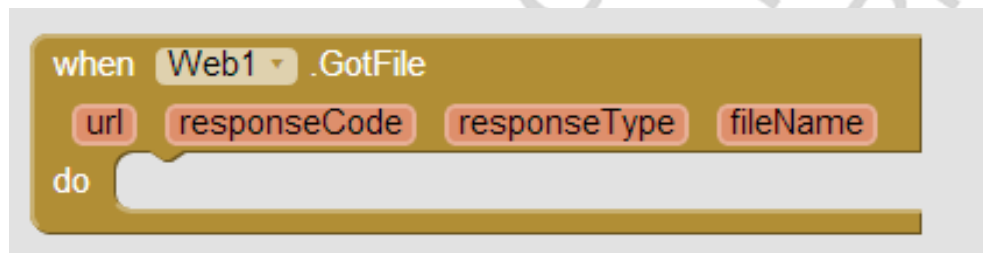
- Url、SaveResponse...

### ■ 事件

- GotFile、GotText.....

### ■ 方法

- PostText、Get.....

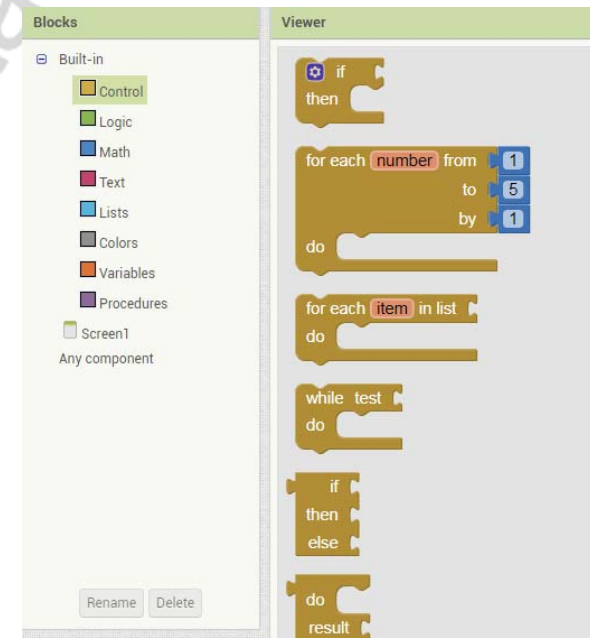
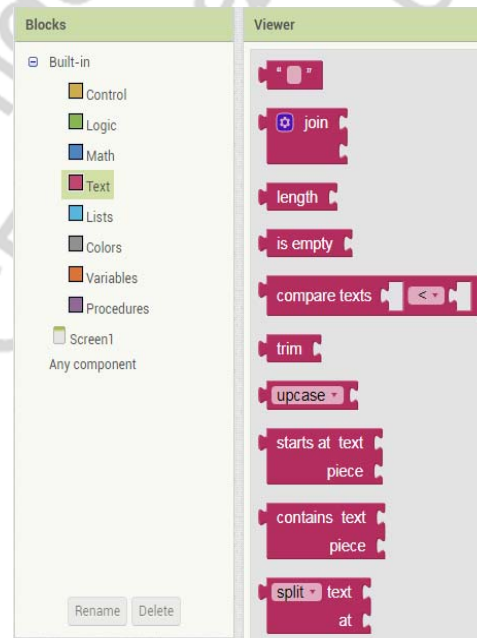
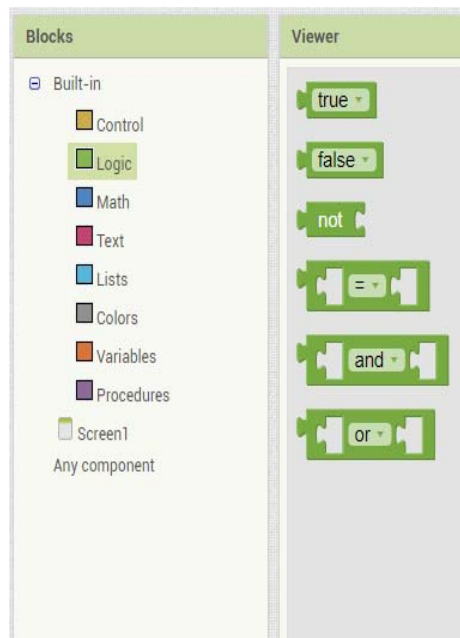




# 附件A – 函式庫

## ■ 各種函式庫

- 運算、程序、顏色
- 更改元件屬性 (標籤的文字大小、顏色、寬度)





# 藍芽協定 – 連接方式

## ■ 事件

- DeviceFound() 、 Connected() 、 BytesReceived()...

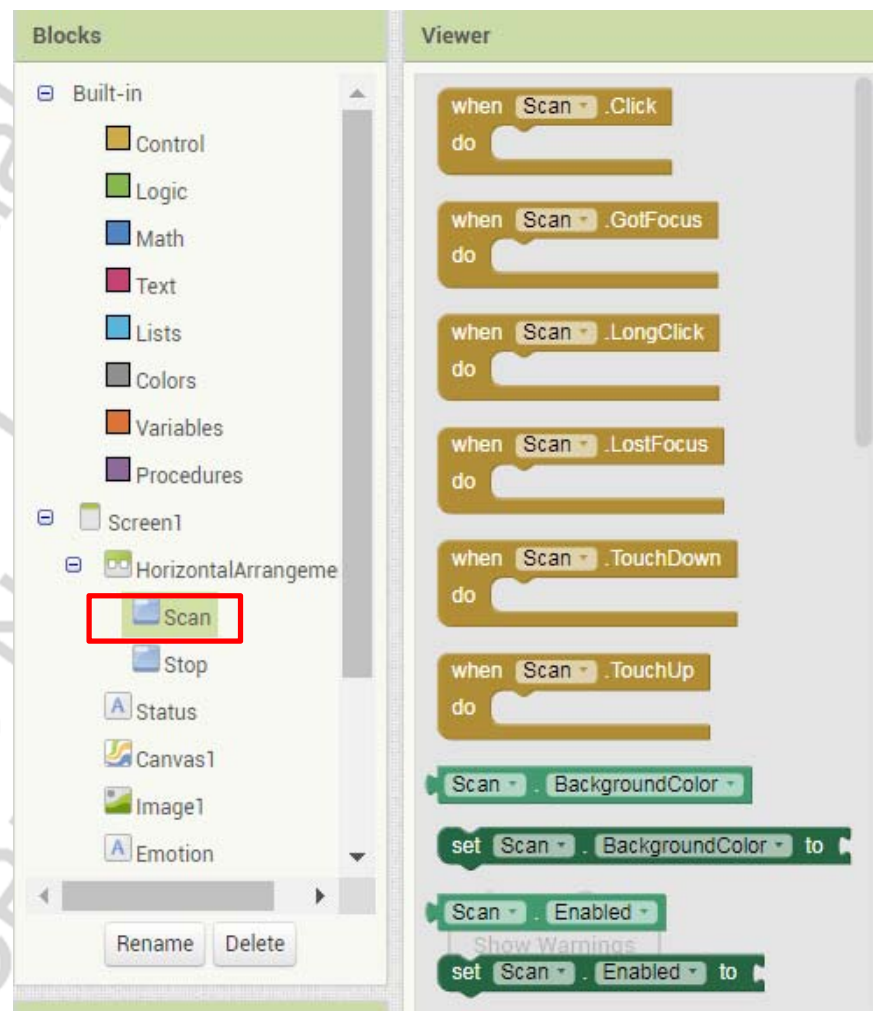
## ■ 方法

- StartScanning() 、 RegisterForBytes() 、 Connect()...



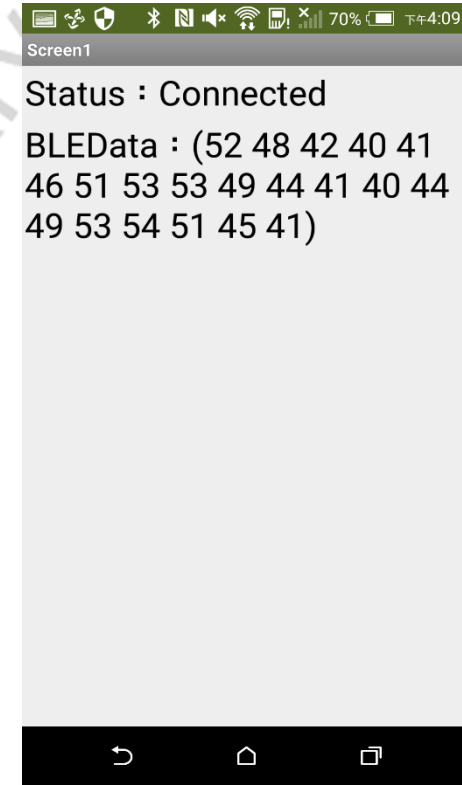
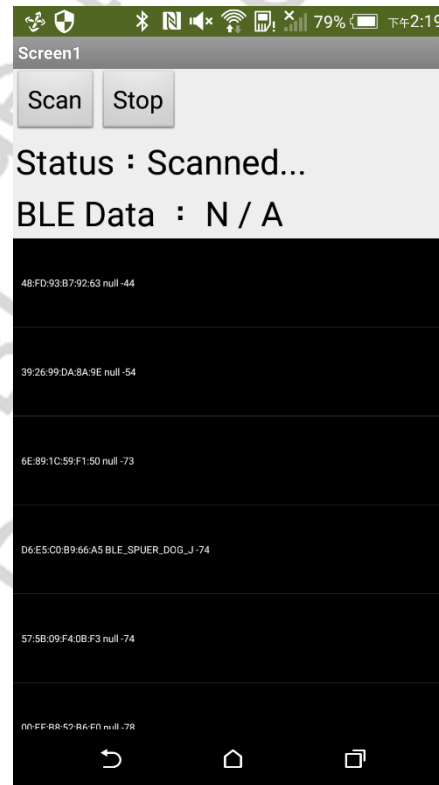
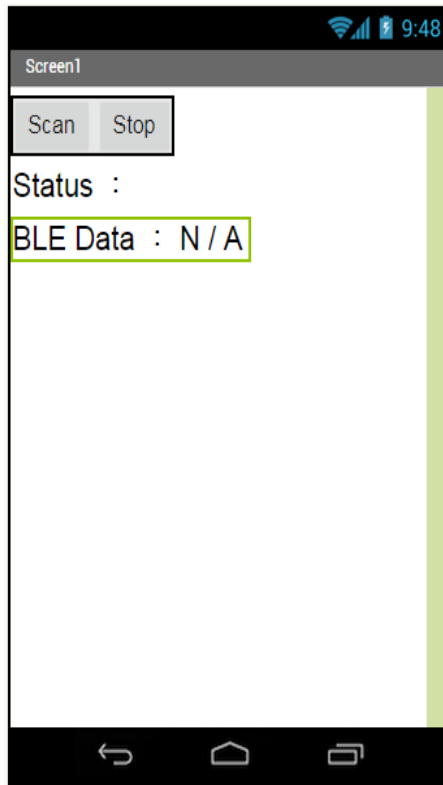
# 實作 – 程式設計

- 內建
  - Control 、 Logic...
- 元件
  - Button 、 Label...



## Lab 2-2

- 實驗目的：實作一個可以連接藍芽裝置的App，透過匯入網路上的open source 檔案，再新增所需要的按鈕與按鈕，完



# Lab 2-2

- 實驗步驟：1. 網路上搜尋 “app inventor ble” → 選擇 “MIT App Inventor Extensions” → 下載 BluetoothLE.aix

Search results for "app inventor ble".

**MIT App Inventor Extensions**

appinventor.mit.edu/extensions

MIT App Inventor Extensions This is the official resource for the MIT App Inventor Extensions, use them within your own projects. Explore, create and share new functionality through App Inventor Extensions. The Supported binaries here have been compiled and ...

**Arduino 101 + App Inventor Ble tutorial - YouTube**

www.youtube.com/watch?v=a1S-abFOZMA

New features in App Inventor BLE with Arduino 101 board Masterad.it European Maker Week 2016 Fabio D'Agnano

**[App Inventor 物聯網應用-藍牙4.0 BLE] Lesson 1 – LED 閃爍 – CAVE...**

blog.cavedu.com/2016/04/09/app-inventor-藍牙4-0-ble-part...

話說從 MIT 回來之後一推沒聞著，今天要介紹的是還沒正式公布的秘辛 – App Inventor 的 BLE (Bluetooth Low Energy) 元件，請用 <http://ble-test.appinventor.mit.edu> 這個測試用伺服器來測試，當然您得先買一片 Arduino 101才行

**[ MediaTek ] 使用 App Inventor 經由 BLE 控制 LinkIt 7697 LED 教學**

oranwind.org/gf-mediatek-shi-yong-app-inventor-kong-zhi...

聯發科於 2017 年 5 月推出 LinkIt 7697，它是一片高度整合的系統單晶片的物聯網應用開發板，特色包含一個 Cortex-M4 微控制器、低功耗 11T1R 802.11b/g/n 2.4GHz Wi-Fi 子系統、低功耗藍牙子系統，以及 PMU，本文描述如何透過 App Inventor 來控制 LED 燈。

**[App Inventor IoT] Lesson 3: Read Arduino 101's analog pin - AppInv...**

www.appinventor.tw/ble\_analogread

App Inventor 中文學習網，期待喜愛 Android 程式設計的朋友們一同分享 搜尋這個協作平台 重要網站 App Inventor 2 來了 ... Android phone (Must have BLE hardware, but I think it's not a problem for most Android devices today). 2.

MIT App Inventor Extensions

This is the official resource for the MIT App Inventor Extensions, use them within your own projects. Explore, create and share new functionality through App Inventor Extensions.

The Supported binaries here have been compiled and tested for release nb170, all others are not supported yet.

Try App Inventor Extensions »

Note: Make sure you have the latest MIT AI2 Companion application. Click on Help in the menubar of your App Inventor screen and select Companion Information to get a QR code or link that can be used to download and install the MITAI2Companion.apk on your phone.

Supported:

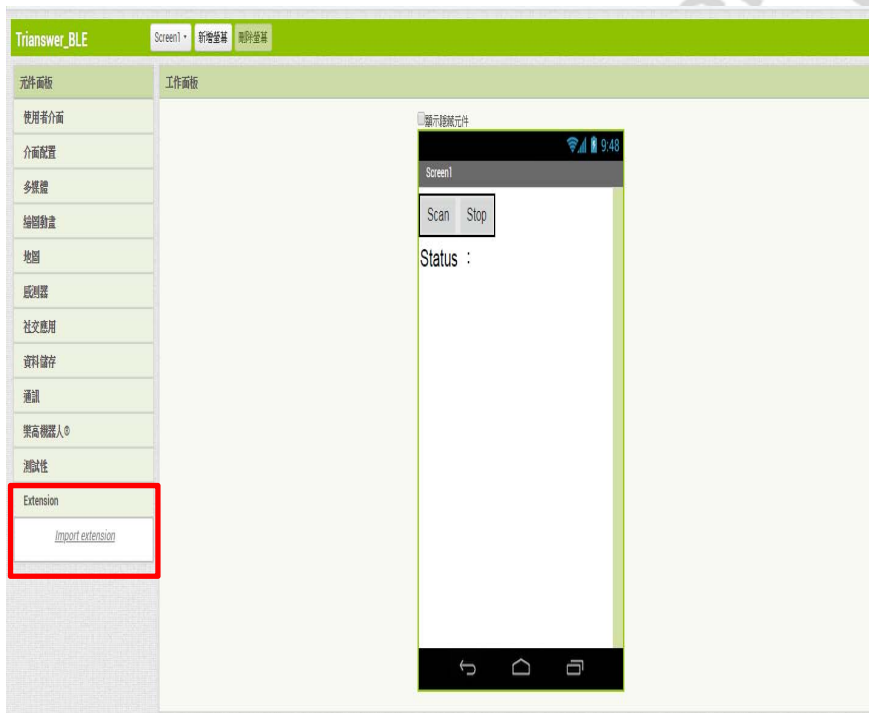
| Name        | Description                                                                                                                        | Author           | Download .aix File | Source Code            |
|-------------|------------------------------------------------------------------------------------------------------------------------------------|------------------|--------------------|------------------------|
| BluetoothLE | Adds as Bluetooth Low Energy functionality to your applications. See BluetoothLE Documentation and Resources for more information. | MIT App Inventor | BluetoothLE.aix    | BluetoothLE-Source.zip |
| Look        | Adds object recognition using a neural network compiled into the extension.                                                        | MIT App Inventor | Look.aix           | Via GitHub             |





## Lab 2-2

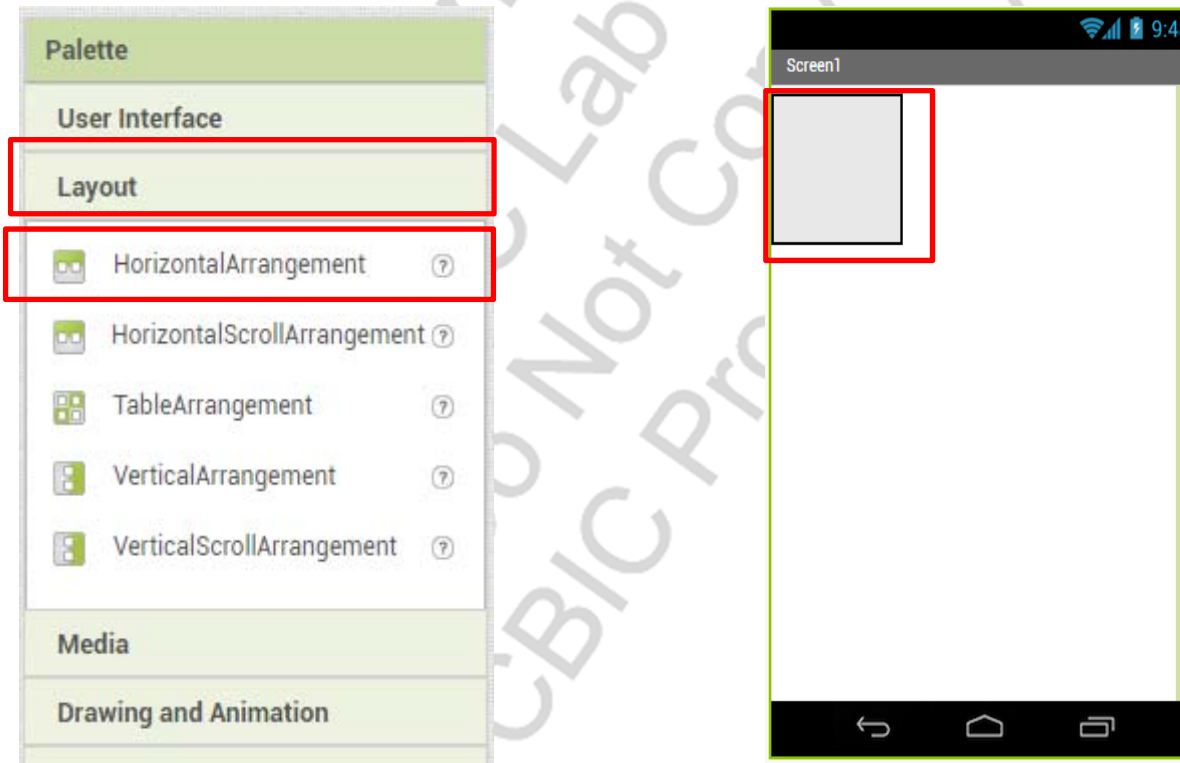
- 實驗步驟：2. 匯入 App inventor2 → 點選元件面板的 extension 的 “import extension” → 選取剛剛下載的檔案上傳





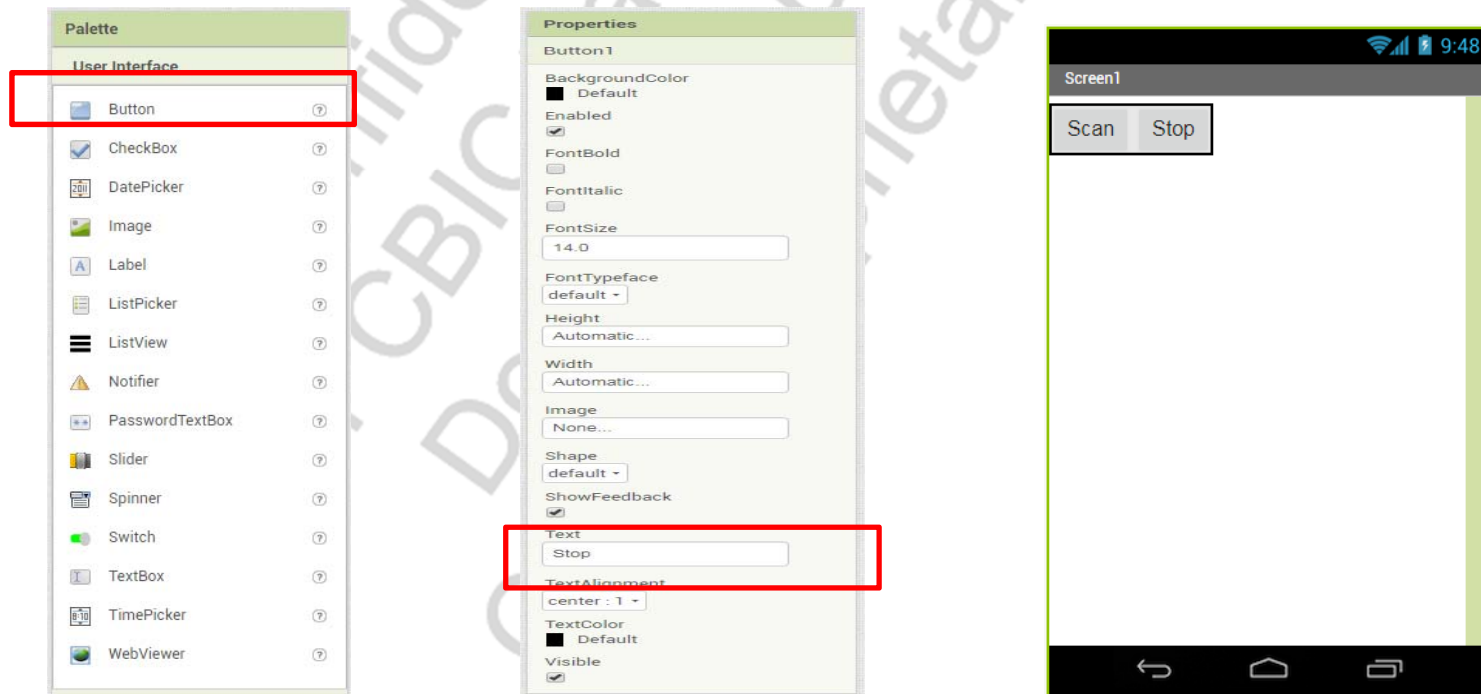
## Lab 2-2

- **實驗步驟：** 3. 加入水平配置方法：從左側的元件面板下方選擇介面配置，選取水平配置，按住滑鼠右鍵**拖曳**至App 螢幕顯示區。



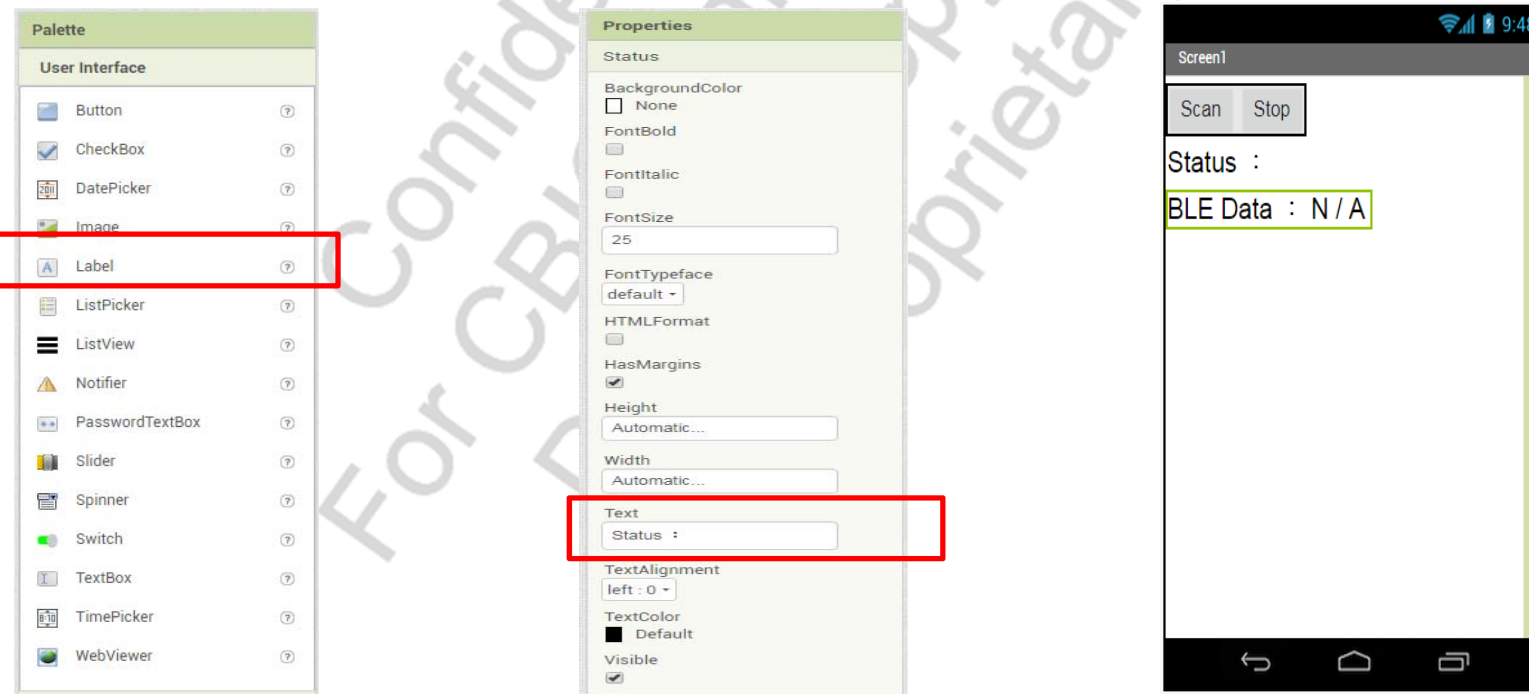
## Lab 2-2

- 實驗步驟：4. 新增按鈕方法：選取**按鈕鍵**，**按住滑鼠右鍵拖曳**至App螢幕顯示區的水平配置區塊內，在右側元件屬性區更改按鈕名稱，依序新增兩個按鈕( Scan 、 Stop)，如右下圖。



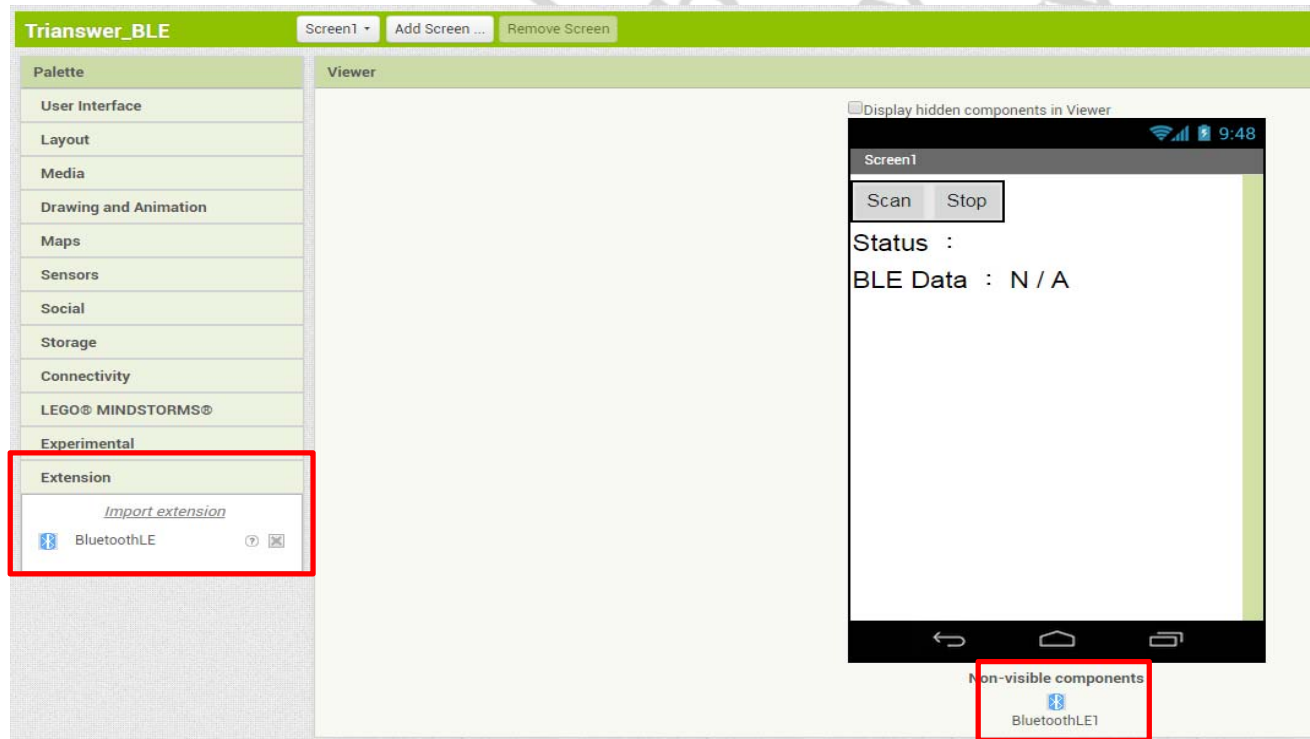
## Lab 2-2

- 實驗步驟：5. 新增標籤方法：選取標籤鍵，按住滑鼠右鍵拖曳至App螢幕顯示區，在右側元件屬性區更改標籤名稱，新增兩個標籤( Status：以及 BLE Data：N / A )，如右下圖



## Lab 2-2

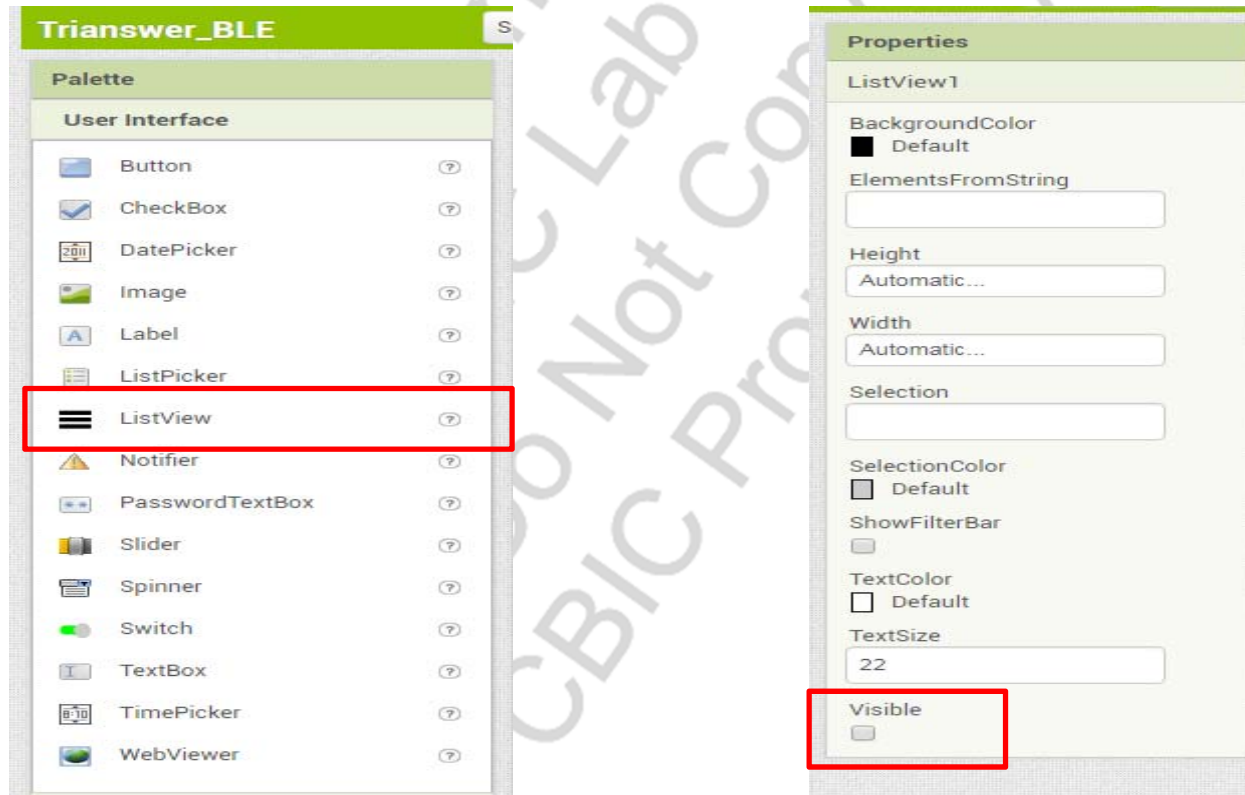
- 實驗步驟：6. 新增藍芽裝置方法：選取左側元件面板的 Extension，選取BluetoothBLE，按住滑鼠右鍵拖曳至App螢幕顯示區。





## Lab 2-2

- **實驗步驟：** 7. 新增清單顯示器方法：選取清單選式器鍵，按住滑鼠右鍵**拖曳**至App螢幕顯示區，並將元件屬性的可見性取消(不要勾起來)。

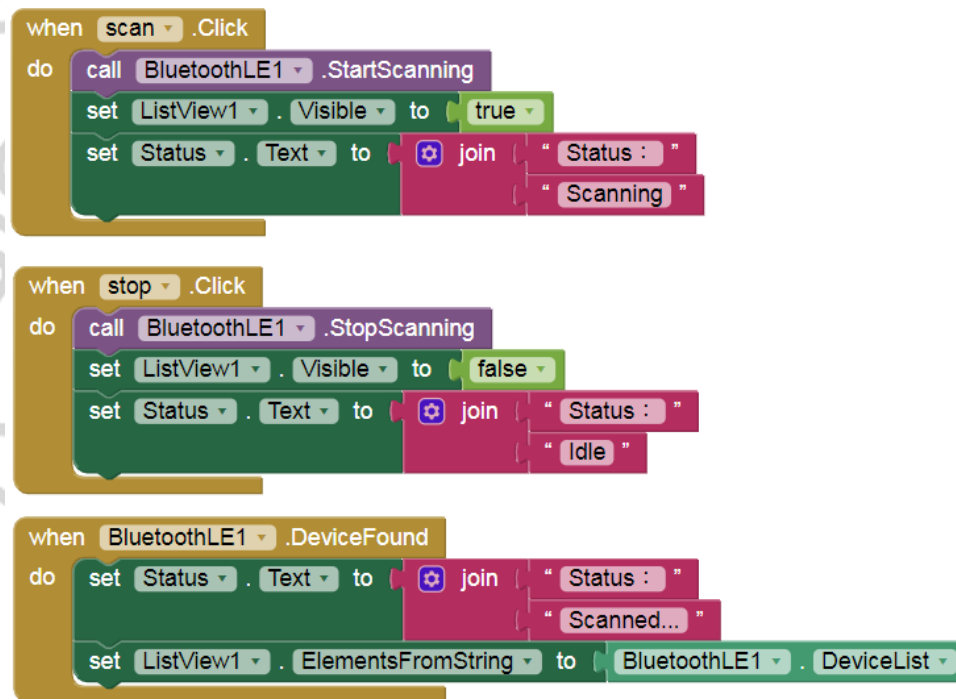
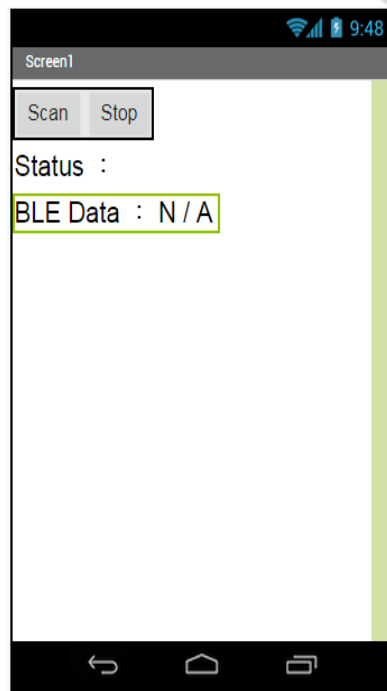






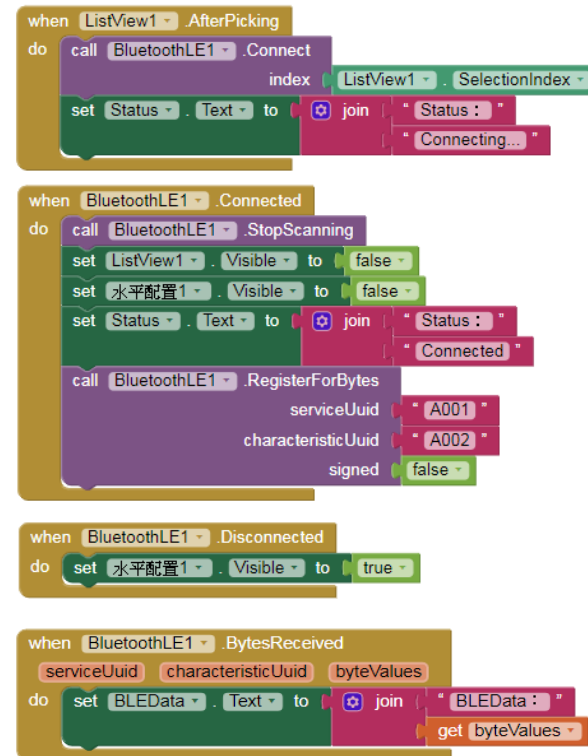
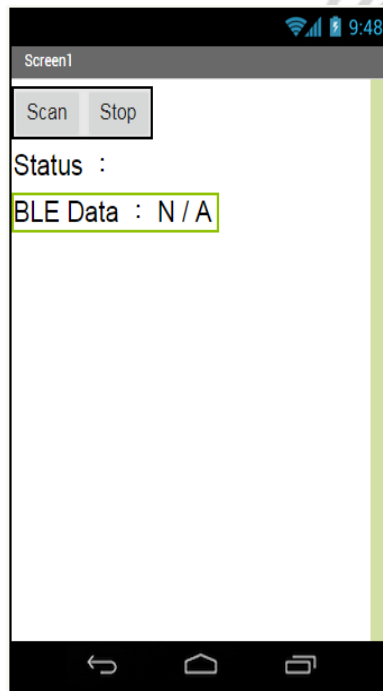
## Lab 2-2

- **實驗步驟：** 8. 進入程式設計頁面，設定**Scan按鈕**搜尋藍芽  
→ 搜尋成功就讓**清單顯示器**出現可見性，接著從清單中選擇要連線的藍芽裝置 → 進行連線，最後設定**Stop按鈕**結束連線。



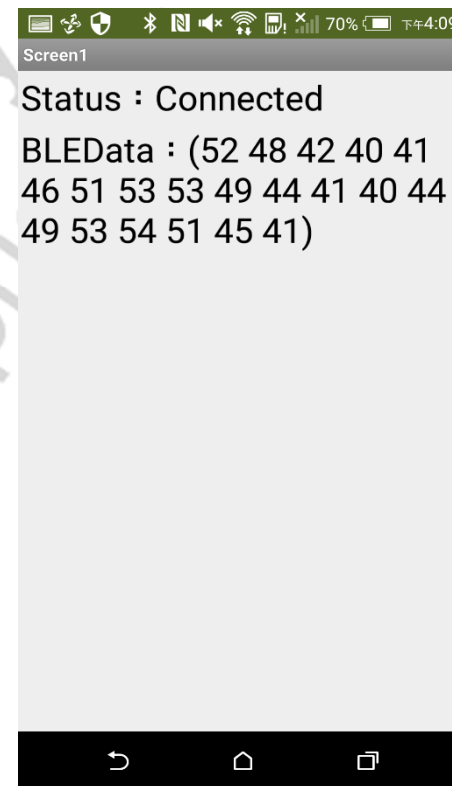
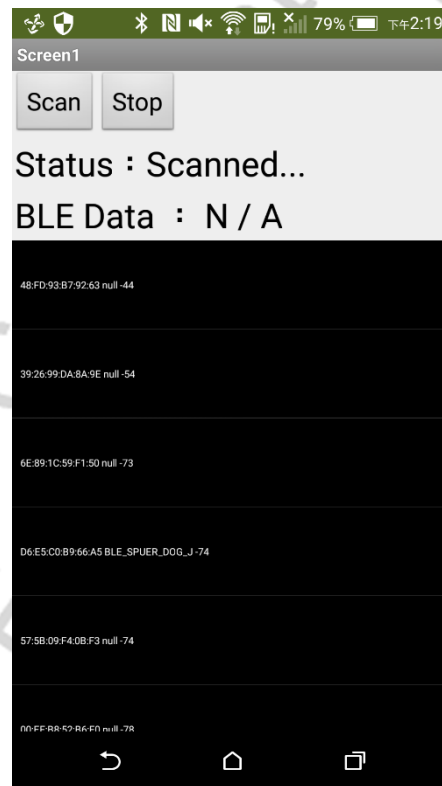
## Lab 2-2

- **實驗步驟：** 9. 此步驟設定兩種狀態，第一種是藍芽成功連線時，需要設定各個元件各自的動作，第二種則為藍芽斷線後，各個元件也有相對應的動作需要被設定。



## Lab 2-2

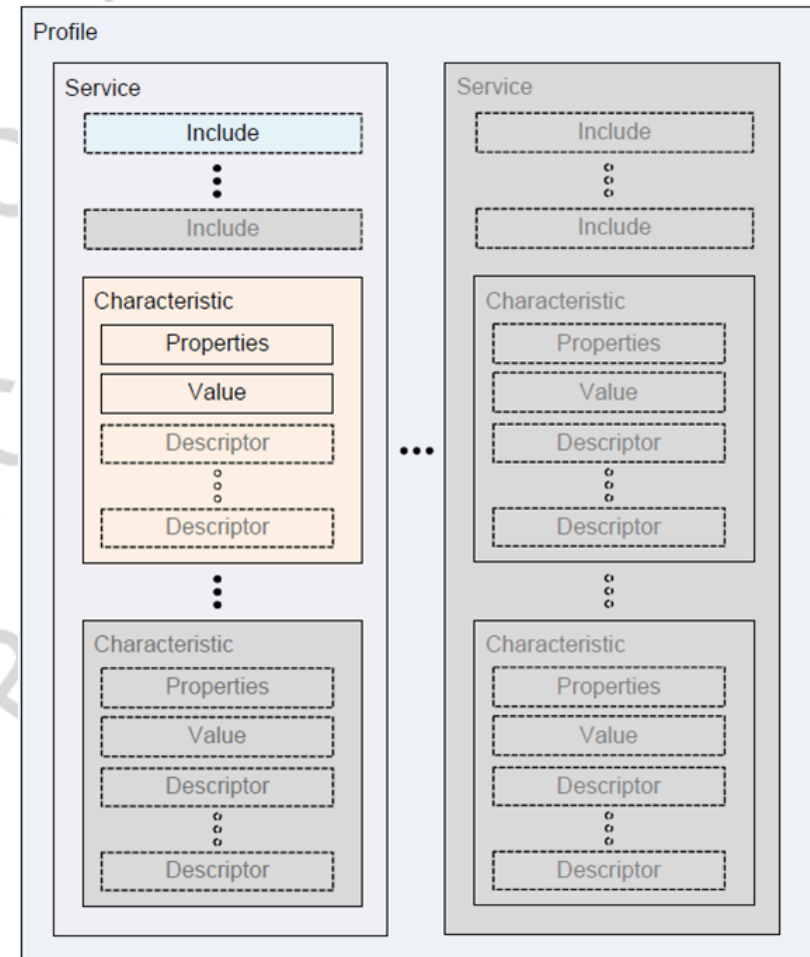
- 實驗步驟：10. 連接手機模擬藍芽功能。





## 附件 B – 藍芽介紹

- 藍芽協定介紹 – GATT
  - Properties - read, write, notify
  - Value – 主要內容
  - Descriptor – 說明





## 附件 B – 藍芽元件

### ■ 事件

- DeviceFound() 、 Connected() 、 BytesReceived()...

### ■ 方法

- StartScanning() 、 RegisterForBytes() 、 Connect()...



# APP實作

## Lab 3 : 應用心律演算法

---







# 心律演算法

- 心跳的長期律動
- 心跳 – bpm





# Tony-Kevin detection

- 相對最大值 ( Local maximum ) 與設定水平以及垂直閾值 ( Threshold )
- 演算法步驟：

1. 找出相對最大值

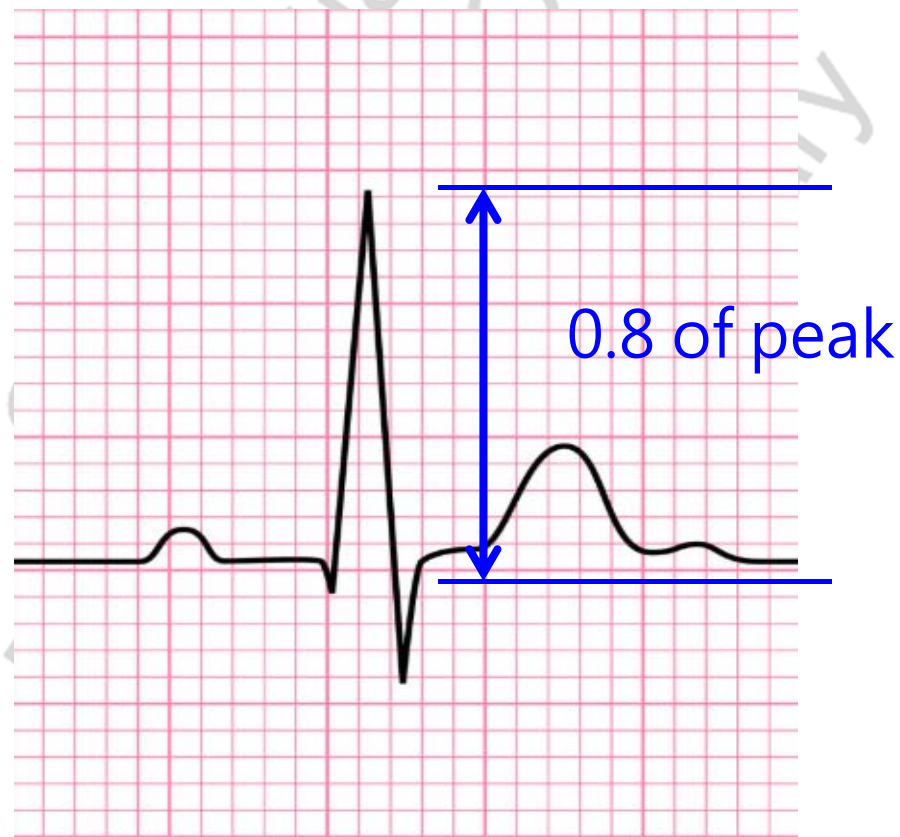




# Tony-Kevin detection

- 演算法步驟：

2. 找出峰值是否有達到設定的垂直閾值

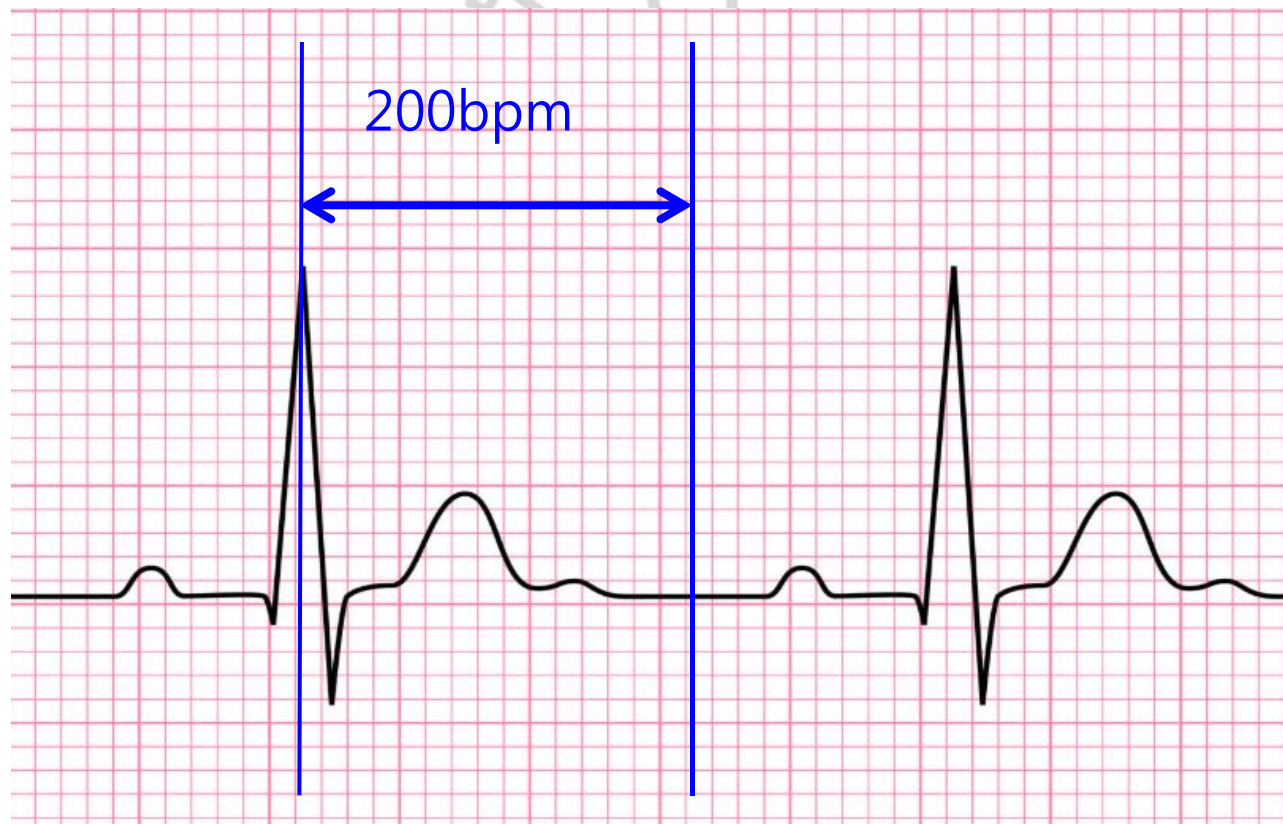




# Tony-Kevin detection

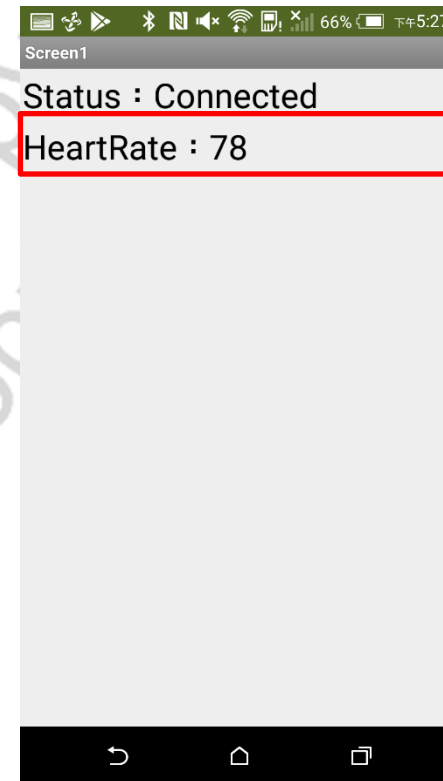
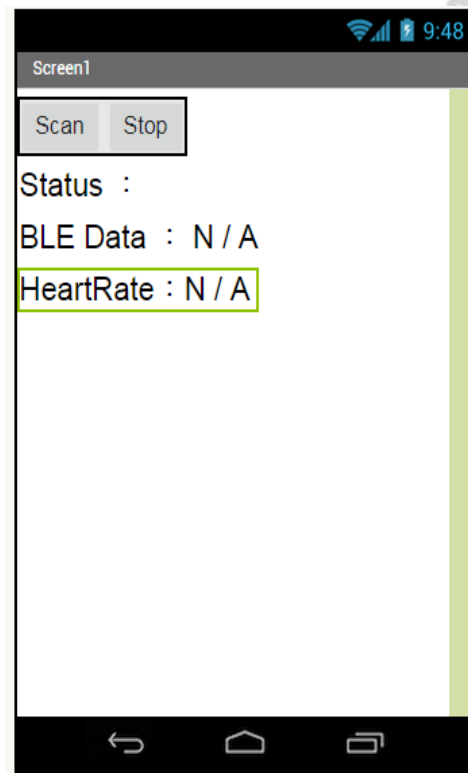
## ■ 演算法步驟：

3. 比較是否時間間隔也有達到水平的閾值



# Lab 3

- 實驗目的：延續Lab2的專案，加入心律演算法，連上藍芽後可以計算出自己的心律。





# Lab 3

## ■ 實驗步驟：

1. 進入程式設計頁面，設定變數用來計算心律，並在計算出心律之後更改其他元件的屬性，讓心律結果顯示在標籤上。

```
initialize global max to -200
initialize global lastPeakindex to 0
initialize global maxindex to 0
initialize global runningPeakindex to 0
initialize global indexThreshold to 200
initialize global waveThreshold to 50
initialize global rri to create empty list
initialize global sampleRate to 500
initialize global list to create empty list

to HeartRateCalculation x x2
do
 initialize local heartrate to (get global sampleRate) x (60 / (get x2 - get x))
 in set HeartRate . Text to join "HeartRate : "
 get heartrate
```

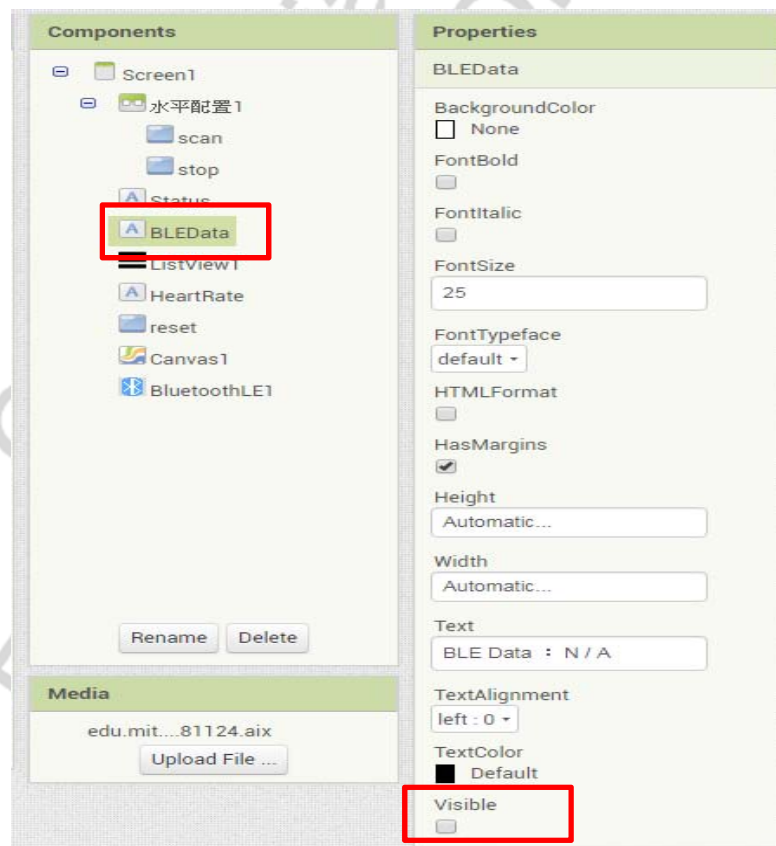
```
to saveData data
do
 append to list list1 get global list
 list2 get data

when BluetoothLE1 . BytesReceived
serviceUuid characteristicUuid byteValues
do
 call saveData
 data get byteValues
 for each (item) in list get byteValues
 do
 if (get global max > get item)
 then
 if (get global max - get item > get global waveThreshold)
 then
 if (get global maxindex > get global indexThreshold)
 then
 call HeartRateCalculation
 x 0
 x2 get global maxindex
 set global runningPeakindex to (get global runningPeakindex - get global maxindex)
 set global maxindex to 0
 else
 set global max to -200
 else
 set global max to get item
 set global maxindex to get global runningPeakindex
 set global runningPeakindex to (get global runningPeakindex + 1)
```



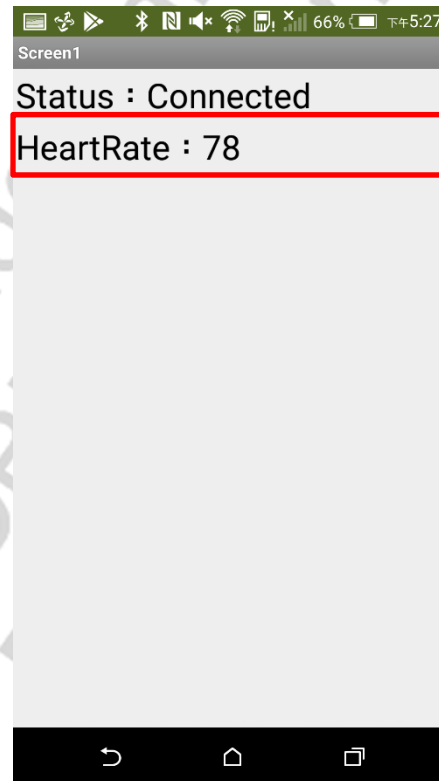
# Lab 3

- 實驗步驟：2. 關閉BLEData標籤的可見性，將勾選取消。



## Lab 3

- 實驗步驟：3. 使用手機模擬，確認是否能成功計算出自己的心律。



# APP設計介紹

## Lab 4 : App Inventor 2 加入波形圖

---



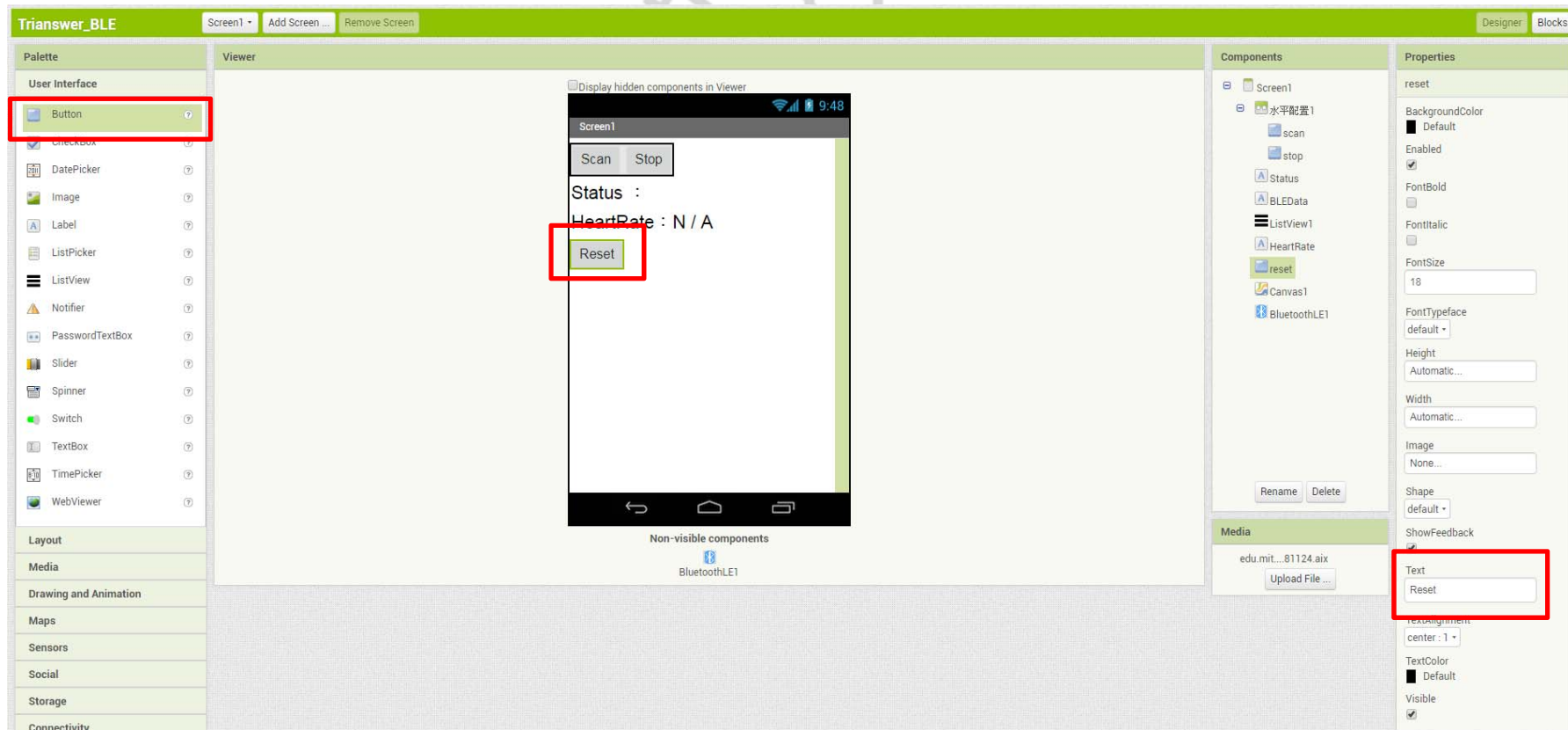
# Lab 4

- 實驗目的：透過藍芽傳送過來的資料，搭配函式後，畫出自己的心電波形圖。



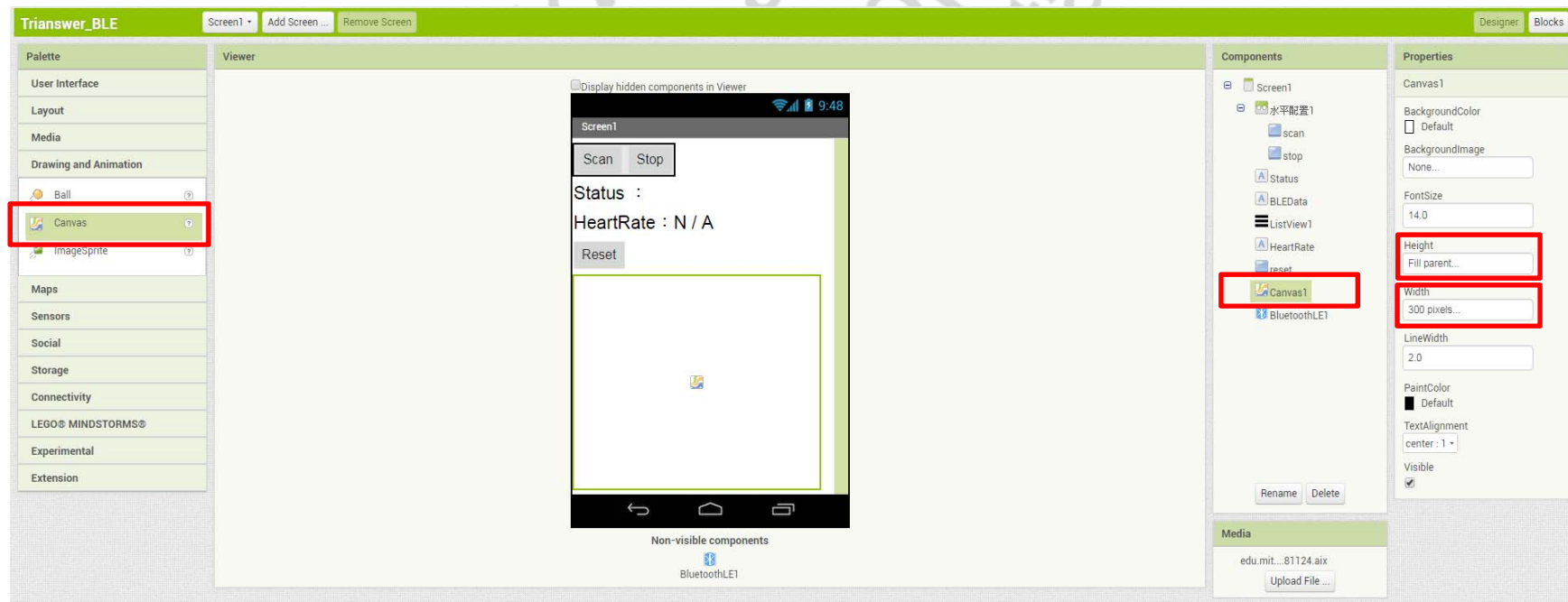
# Lab 4

- 實驗步驟：1. 新增一個按鈕，命名為 reset，用來初始化各項變數。



# Lab 4

- 實驗步驟：2. 新增畫布方法：選擇左側元件面板的繪圖動畫，選擇畫布，按住滑鼠右鍵拖曳至螢幕顯示區，命名為Canvas1，高度設為填滿，寬度設為300像素。





# Lab 4

## ■ 實驗步驟：

3. 進入程式語言頁面，設定各個變數，以及接收資料後進行畫圖的動作。

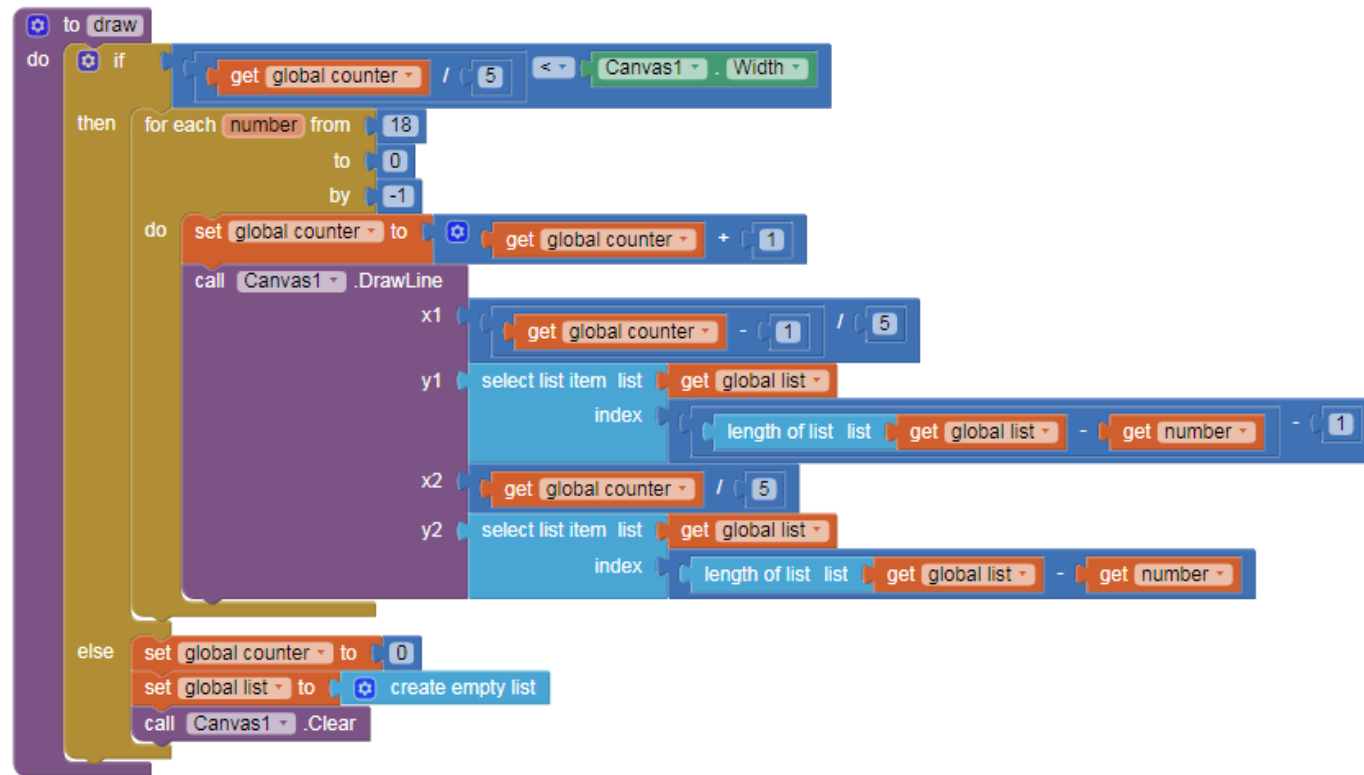
```
initialize global counter to 0

when reset.Click
do
 set global counter to 0
 set global lastPeakindex to 0
 set global max to -200
 set global maxindex to 0
 set global runningPeakindex to 0
 set global list to create empty list
 set global rri to create empty list
 call Canvas1.Clear
```

```
when BluetoothLE1.BytesReceived
 serviceUuid characteristicUuid byteValues
do
 call saveData
 data get byteValues
 call draw
 for each item in list get byteValues
 do
 if get global max > get item
 then
 if get global max - get item > get global waveThreshold
 then
 if get global maxindex > get global indexThreshold
 then
 call HeartRateCalculation
 x 0
 x2 get global maxindex
 set global runningPeakindex to get global runningPeakindex - get global maxindex
 set global maxindex to 0
 else
 set global max to -200
 else
 set global max to get item
 set global maxindex to get global runningPeakindex
 set global runningPeakindex to get global runningPeakindex + 1
```

# Lab 4

- 實驗步驟：4. 定義程序 draw ，當程式呼叫此程序時，執行以下動作，完成繪圖這項功能。



# Lab 4

- 實驗步驟：4. 使用手機模擬，確認App可以成功畫出波形。

