# EE3014 智慧物聯感測與實作 Lab 3 實驗報告

組別:1

組員:

[學號 B1121149] [姓名:張嘉宸]

[學號 B1121141] [姓名:葉彥辰]

# 一、 簡介

請說明實作(驗)之項目內容與學習目的(字體:新細明體,大小: 14,單行間距,左右對齊)

這次實作透過 App inventor 2 來派列程式方塊,使我們能夠讀取手機中加速規的方向軸和大小,再結合時脈輸出單軸重力加速度大小對應時間的訊號。

#### 二、步驟

若為 TriAnswer 平台或是慣性運動感測平台之實作,請配合以 流程圖與照片或是螢幕截圖,分別詳細說明硬體與軟體之操作 步驟。

若為 App Inventor 2 之實作,請依 Designer view 與 Block view 詳細說明各元件與程式區塊之設計,佐以 App Inventor 2 之螢幕截圖來說明。

(字體:新細明體,大小:14,單行間距,左右對齊) Lab3\_1

```
when Screen1 * Initialize

do set StartButton * . Enabled * to true *

set EndButton * . Enabled * to false *

set xText * . Text * to 0

set yText * . Text * to 0

set Phi3Text * . Text * to 0

set Phi1Text * . Text * to 0

set Phi1Text * . Text * to 0

set AccelerometerSensor1 * . Enabled * to false *

when StartButton * . Click

do set StartButton * . Enabled * to false *

set AccelerometerSensor1 * . Enabled * to true *

set EndButton * . Enabled * to true *
```

這部分的功能是當螢幕被使用被激活時,將所有參數先歸零初始 化,並指保留 START 鍵可按,END 則設定為 FALSE 來讓他反白, 也關閉加速規功能。而在 START 鍵被按下後,START 鍵便反白無 功能,變成 END 可以停止功能,也同時啟用加速規使其賦能。

```
when AccelerometerSensori AccelerationChanged

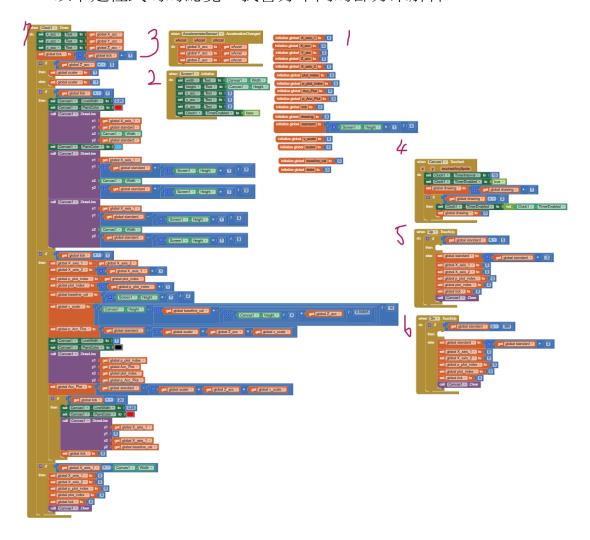
XAccel yAccel zAccel

do set XText 1 to get xAccel pet yAccel pet zAccel pet yAccel p
```

這部分的功能是讓三軸的 TEXTBOX 去抓加速規的值,並套入 arctan 的公式求出 phi,以及當 END 被按下後保留 START 鍵可按, END 則設定為 FALSE 來讓他反白,並將所有參數先歸零初始化設 為 0。並且關閉加速規的偵測。

# Lab3 2

以下是程式碼的總覽,我會分不同的部分來解釋。



# 第一部分



上圖就只是一大坨的變數初始化。

# 第二部分

```
when Screen1 v .Initialize

do set width v . Text v to Canvas1 v . Width v set height v . Text v to Canvas1 v . Height v set x_acc v . Text v to 0 set y_acc v . Text v to 0 set z_acc v . Text v to 0 set Clock1 v . TimerEnabled v to false v
```

當螢幕被開啟的時候,第一件做的事情就是這個,把不同的變數另 成初始值。

## 第三部分

```
when AccelerometerSensor1 · .AccelerationChanged

xAccel yAccel zAccel

do set global X_acc · to ( get xAccel · get yAccel · get global Y_acc · to ( get yAccel · get zAccel ·
```

主要是在講說,當加速度規的值改變的時候,就要持續地將此時加速度規的變數存到全域變數裡面,才可以對他實施操作。

第四部份

```
when Canvas1 v .Touched

x y touchedAnySprite

do set Clock1 v . TimerInterval v to 10

set Clock1 v . TimerEnabled v to true v

set global drawing v to get global drawing v + 11

then set Clock1 v . TimerEnabled v to not Clock1 v . TimerEnabled v to set global drawing v to 0
```

當螢幕被按下的時候,drawing 這個變數的值會在0與1之間交替。若 drawing 為1,則 Clock.TimeEnabled 的值會設為 True,反之則設為 False,這會影響到之後的第七部分。

第五部分及第六部分

```
when Up .TouchUp
do
    🔯 if
                get global standard *
    then
    else
          set global standard v to
                                  get global standard •
                                                             + [ -3
          set global X_axis_1 v to 0
          set global X_axis_2 to 0
          set global p_plot_index v to 0
          set global plot_index v to 0
          set global tick v to 0
          call Canvas1 .Clear
 when Dn 🔻
            .TouchUp
do
     if
                 get global standard *
                                            366
     then
     else
           set global standard v to
                                   get global standard 🔻
           set global X_axis_1 ▼ to 0
           set global X_axis_2 v to 0
           set global p_plot_index v to 0
           set global plot_index to 0
           set global tick to 0
           call Canvas1 .Clear
```

這兩個部分可以一起講,因為他們做類似的事情,都是在調整基準線的位置。而如果是往上調(Up)則會先判斷有沒有高過範圍,若無則實施變高,否則不予理會;反之,下調也是做一樣的事情,先判判斷是否要下調,再下調。

第七部分 上 (第一幀)

```
when Clock1 - .Timer
                                                                    when AccelerometerSensor1 - AccelerationChan
    set x_acc . Text to get global X_acc
                                                                     xAccel yAccel zAccel
     set y_acc . Text to
                             get global Y_acc
     set z_acc . Text to get global Z_acc .
                                                                    do set global X_acc to get xAccel
                                                                        set global Y_acc v to ( get yAccel v
                            get global tick •
                                                                        set global Z_acc v to get zAccel v
                get global Z_acc - < - 0
          set global scaler v to [1]
                                                                   when Screen1 .Initialize
                                                                      set width . Text . to Canvas1 .
          set global scaler to [-1]
                                                                       set height . Text to Canvas1
     🤨 if
                                                                       set x_acc . Text to 0
              get global tick = = 1
                                                                       set y_acc . Text to 0
           set Canvas1 · LineWidth · to 2.25
                                                                       set z_acc · . Text · to 0
           set Canvas1 . PaintColor .
                                                                       set Clock1 · . TimerEnabled · to false
           call Canvas1 .DrawLine
                                   get global X_axis_1 -
                                   get global standard -
                              y1
                                   Canvas1 - Width -
                                   get global standard
           set Canvas1 . PaintColor to
           call Canvas1 .DrawLine
                                   get global X_axis_1 •
                                        get global standard *
                                                                 Screen1 · Height · × 11 / 8
                                   Canvas1 - Width -
                                   get global standard 🔻
                                                                                              1 / (8
                                                                     Screen1 . Height . ×
           call Canvas1 ▼ .DrawLine
                                   get global X_axis_1 ▼
                                     get global standard -
                                                              Screen1 · Height · × 1 / 8
      8 0
                                   Canvas1 - Width -
                              x2 (
                                     get global standard 🔻
w Warnings
                                                              Screen1 - Height - ×
                                                                                          1
```

當 Timer 被打開的時候,就代表要 tick 變數要開始跑了,tick 變數有點像是幀數的概念,就每一個 timer,就是一幀。只有第一幀的時候, 畫基準線那些,在其他幀的時候保留。

第七部分 之二(第一幀以後)

```
then set global x axis 1 to get global X axis 2 set global x axis 1 to get global x axis 1
```

只要幀數大於等於 1,代表我已經畫完基底了(即便是第一幀,也在前面畫完了),這裡就只是單純的描點,方法是有兩個時間軸,x1 跟 x2,代表這一瞬間跟下一瞬間,我們將這一瞬間跟下一瞬間的量測數值連起來之後,將 x1 與 x2 各加一,等待下一幀的繪畫。

第七部分\_之三 (畫細部時間軸)

每20幀,就加一條粉紅色的時間軸。

第七部分 之四(超過)

```
then set global X_axis_1 v to 0
set global X_axis_2 v to 0
set global pplot_index v to 0
set global plot_index v to 0
set global fick v to 0

X 0 call Canvas1 v Clear
```

若全部的 canvas 畫到底了,則就重新歸零到第零幀,等到待會的第一幀去畫基底跟描點。其實也不用等待,因為啟動第一個動作就是tick += 1,因此也不算要等第一幀。

# 結果

若為 TriAnswer 平台或是慣性運動感測平台之實作,請配合量 測波形之截圖說明,並提供所紀錄之量測資料進行所需之結果 分析(請提供資料以及分析作圖之 excel 檔)。

若為 App Inventor 2 之實作,請附上在開發環境中,使用 AI Companion 掃描 QR code 之下載 progresses bar 之截圖,以及 Android/iOS 行動裝置中,app 之操作截圖,並說明操作步驟。

(字體:新細明體,大小:14,單行間距,左右對齊)

Lab3\_1



# Accelerometer Data

 Start
 End

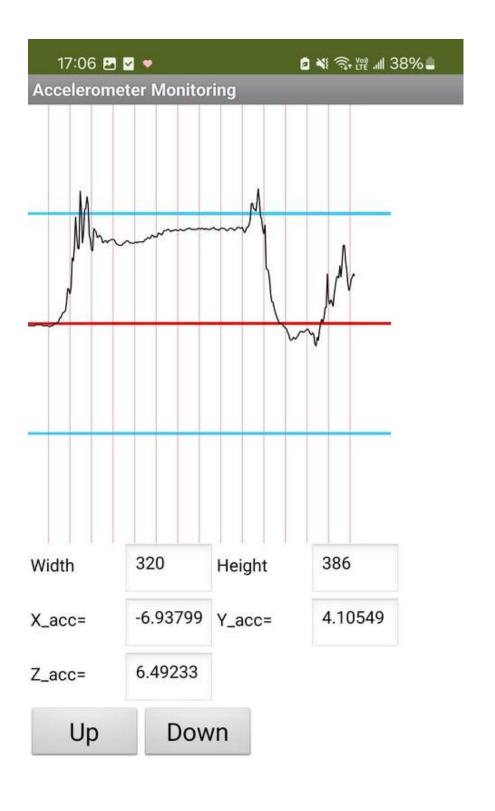
 X= 1.11565
 Phi1= 6.557

 Y= 7.05404
 Phi2= 46.22065

 Z= 6.667
 Phi3= 46.96895

III O <

Lab3\_2



## 三、 討論與心得

請討論於此實作(驗)中,所遇到之問題,與學習心得。

(字體:新細明體,大小:14,單行間距,左右對齊)

這次實作有兩部分,分別為偵測加速貴的方位大小和角度,以 及將單軸的加速度大小隨時間推進繪製成訊號圖。第一部分很 迅速地就完成了,運用到的只是基本的賦值和 arctan 的計算, 透過拉方塊來組合出公式算出 phi 也就是角度。第二部份我們 卡關在繪製背景線條的部分,我們在設定完初始值和繪製訊號 圖背景的方塊後就先行測試,只是好幾次都沒顯示出來,在經 歷好幾次修改後終於做出功能正常的 app,這次用 app inventor 2 設計出的程式和上次用 tri-answer 手機平台 BLE 傳輸的 ECG,PPG 訊號很類似,或許這個程式能用在很多生物訊號的 繪製上。