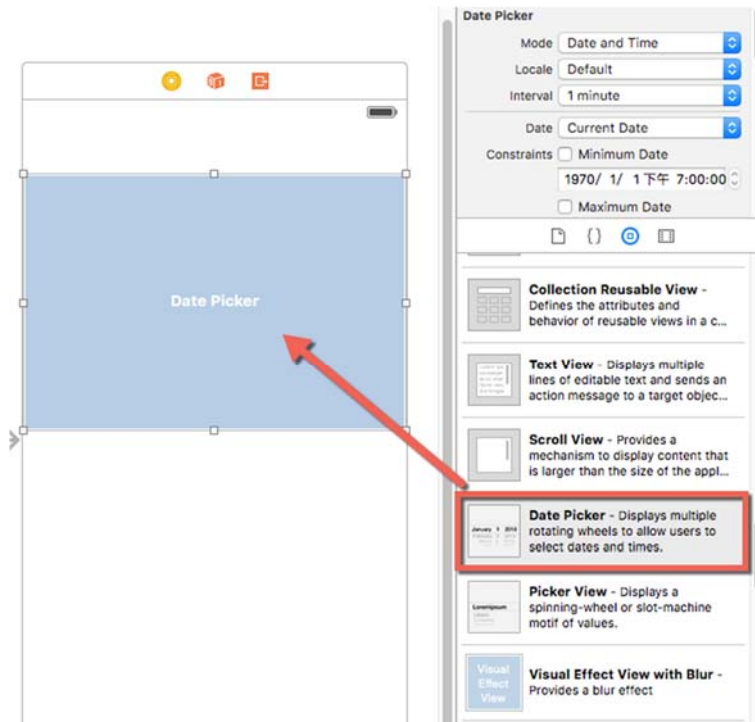


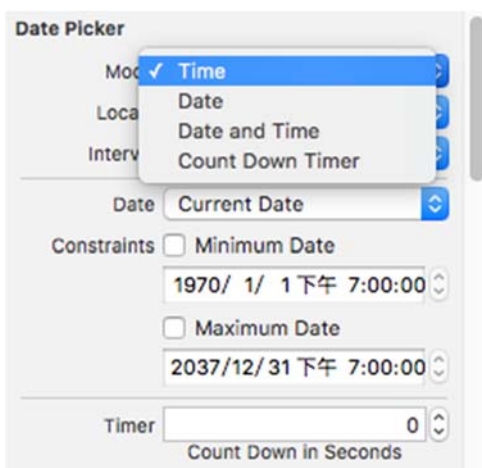
6-5

解答

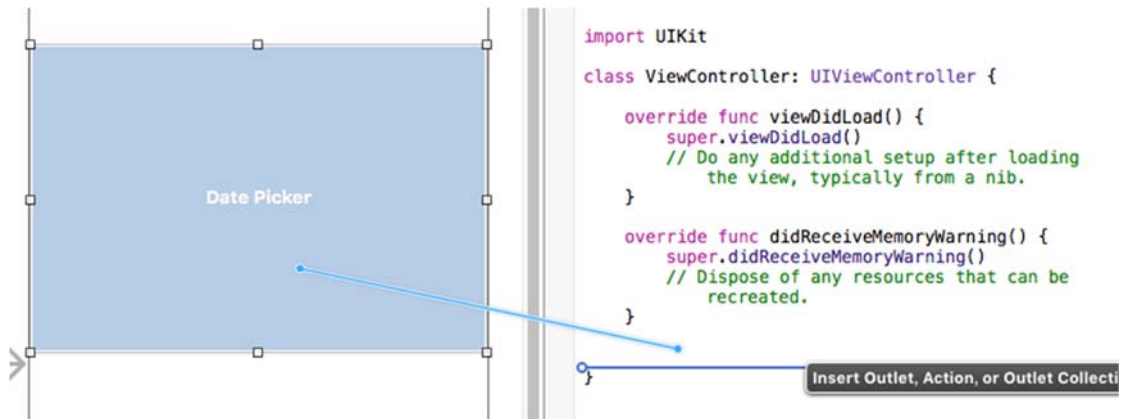
1. 拖曳【Date Picker】建立在 IB 畫面中。



2. 在【Date Picker】元件的「屬性檢視」的視窗，Mode 中設定顯示時間 Time 部分就好



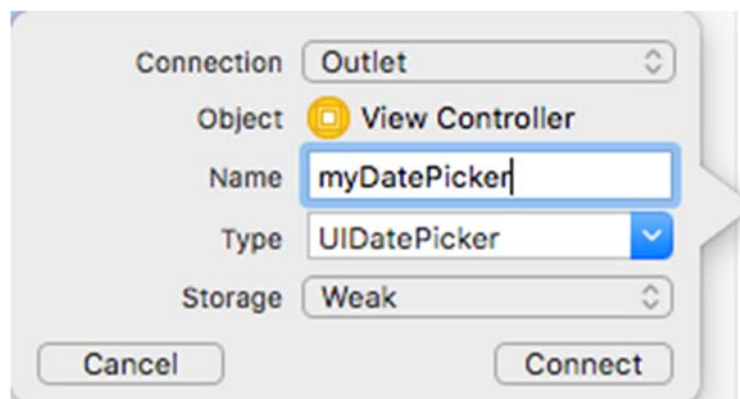
3. 在 IB 視窗上點選【Date Picker】，接著按住【control】鍵，用滑鼠拖曳到右邊視窗與【程式碼】建立連結。



按住【Control】用滑鼠拖曳【Date Picker】元件。

4. 由【物件區】選取之元件，設定名稱如下：

在「Connection」欄位點選【Outlet】、「Name」欄位設定名稱【myDatePicker】以及在「Type」欄位點選【UIDatePicker】後，按【Connect】按鈕。



在拖曳後自動彈跳的視窗

5. 最後，我們在【紅框】中設定 swift 的程式碼在編輯區當中。

```

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically
        from a nib.

        NSTimer.scheduledTimerWithTimeInterval(0.05, target: self,
            selector: ("checkTime"), userInfo: nil, repeats: true)

    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    @IBOutlet weak var myDatePicker: UIDatePicker!

    func checkTime () {
        let now = NSDate()
        let df = NSDateFormatter()
        df.dateFormat = "HH:mm"
        myDatePicker.date = now
    }
}

```

手動輸入紅框內的程式

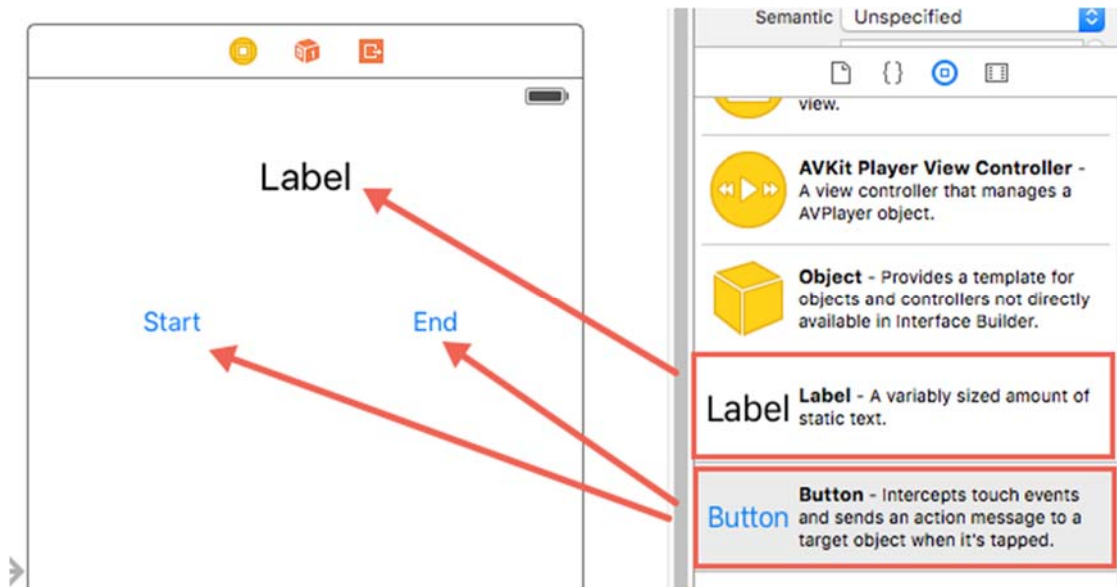
6. 在啟動「RUN」後，每隔一分鐘會發現分鐘的數值會改變。

iPhone 5s - iPhone 5s / iOS 9.2 (13C75)		
Carrier	4:51 PM	
1	48	
2	49	
3	50	AM
4	51	PM
5	52	
6	53	
7	54	

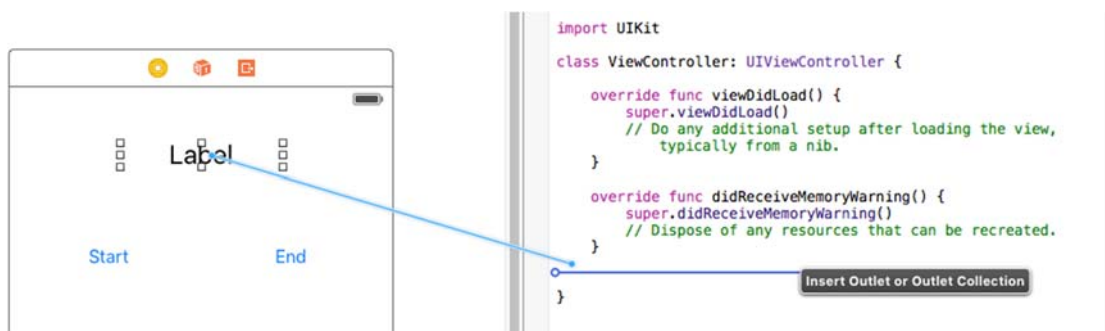
6-6

解答

1. 拖曳【Label】、【Button】建立在 IB 畫面中，並將 Button 中改為 Start 與 End。



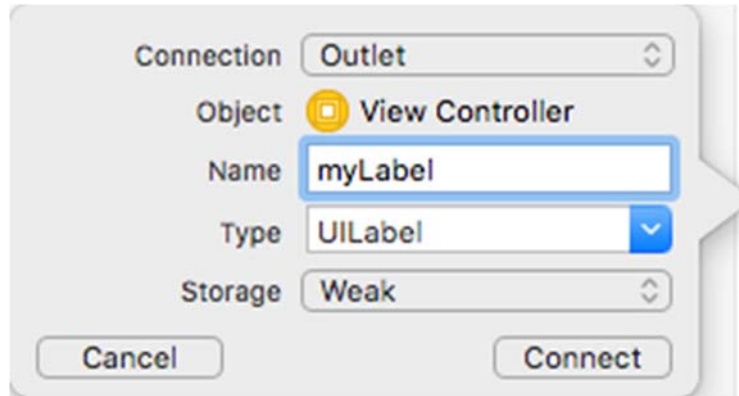
2. 在 IB 視窗上點選【Label】，接著按住【control】鍵，用滑鼠拖曳到右邊視窗與【程式碼】建立連結。



按住【Control】用滑鼠拖曳【Label】元件。

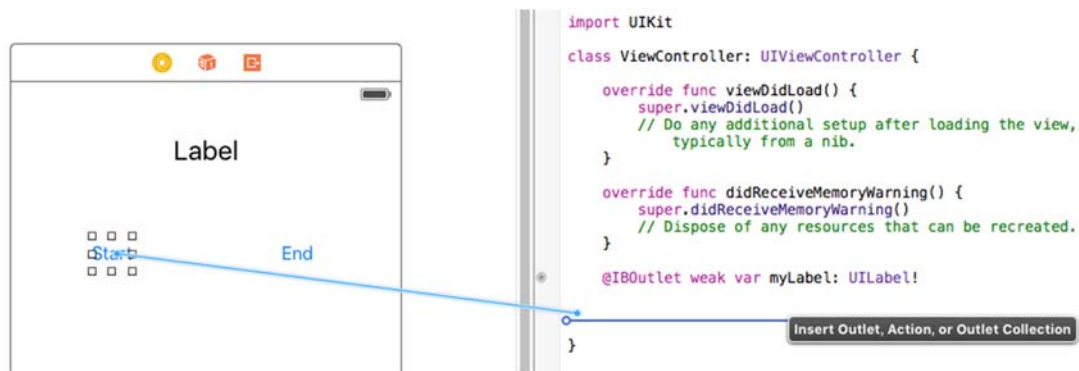
3. 由【物件區】選取之元件，設定名稱如下：

在「Connection」欄位點選【Outlet】、「Name」欄位設定名稱【myLabel】以及在「Type」欄位點選【UILabel】後，按【Connect】按鈕。



在拖曳後自動彈跳的視窗

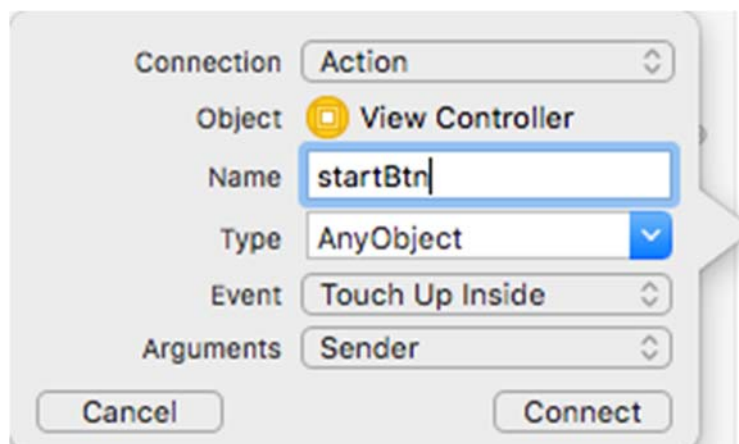
4. 在 IB 視窗上點選 **Start** 【Button】，接著按住 **【control】** 鍵，用滑鼠拖曳到右邊視窗與 **【程式碼】** 建立連結。



按住 **【Control】** 用滑鼠拖曳 **Start** 【Button】 元件。

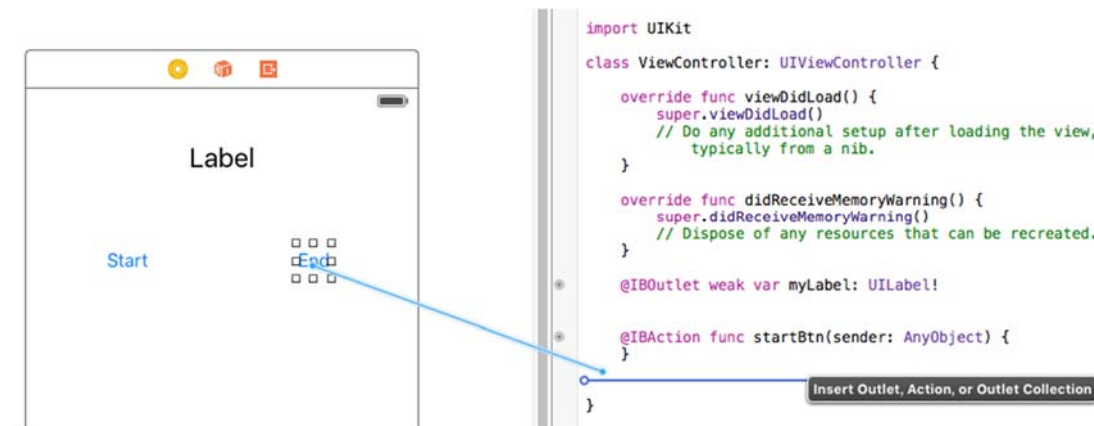
5. 由 **【物件區】** 選取之元件，設定名稱如下：

在「**Connection**」欄位點選 **【Action】**、「**Name**」欄位設定名稱 **【startBtn】**，按 **【Connect】** 按鈕。



在拖曳後自動彈跳的視窗

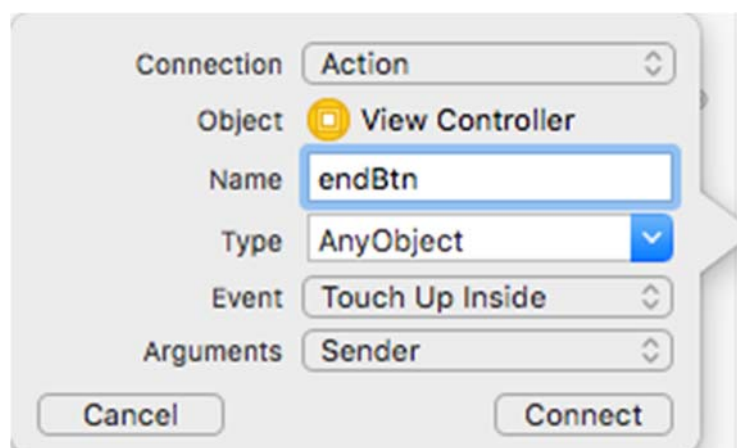
6. 在 IB 視窗上點選 End【Button】，接著按住【control】鍵，用滑鼠拖曳到右邊視窗與【程式碼】建立連結。



按住【Control】用滑鼠拖曳 End【Button】元件。

7. 由【物件區】選取之元件，設定名稱如下：

在「Connection」欄位點選【Action】、「Name」欄位設定名稱【endBtn】，按【Connect】按鈕。



在拖曳後自動彈跳的視窗

8. 最後，我們在【紅框】中設定 swift 的程式碼在編輯區當中。

手動輸入紅框內的程式


```

import UIKit

var startTime = NSDate.timeIntervalSinceReferenceDate()
var timer = NSTimer()

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    @IBOutlet weak var myLabel: UILabel!

    @IBAction func startBtn(sender: AnyObject) {

        startTime = NSDate.timeIntervalSinceReferenceDate()

        timer = NSTimer.scheduledTimerWithTimeInterval(0.01, target: self,
        selector: Selector("updateTime"), userInfo: nil, repeats: true)

    }

    @IBAction func endBtn(sender: AnyObject) {

        timer.invalidate()

    }

    func updateTime () {
        let currentTime = NSDate.timeIntervalSinceReferenceDate()

        let elapsedTime = currentTime - startTime
        let floatelapsedTime = Float (elapsedTime) // 浮點時間

        let minutes = Int16(elapsedTime / 60.0)
        let seconds = Int16(elapsedTime % 60)

        let intelapedTime = Int16(elapsedTime) //整數時間
        let floateIntlapedTime = Float (intelapedTime) // 浮點 整數時間

        //浮點時間 - 浮點整數時間 留下小數位， 乘上100後 再取整數， 就可以得到 毫秒
        let mseconds = Int16( (floatelapsedTime - floateIntlapedTime) * 100 )

        let strMinutes = String(format: "%02d", minutes)
        let strSeconds = String(format: "%02d", seconds)
        let strMSeconds = String(format: "%02d", mseconds)

        myLabel.text = "\(strMinutes):\(strSeconds):\(strMSeconds)"

    }

}

```

7. 在啟動「RUN」後，按下 Start 碼錶就開始計時，按下 end 就停止計時。

00:02:93

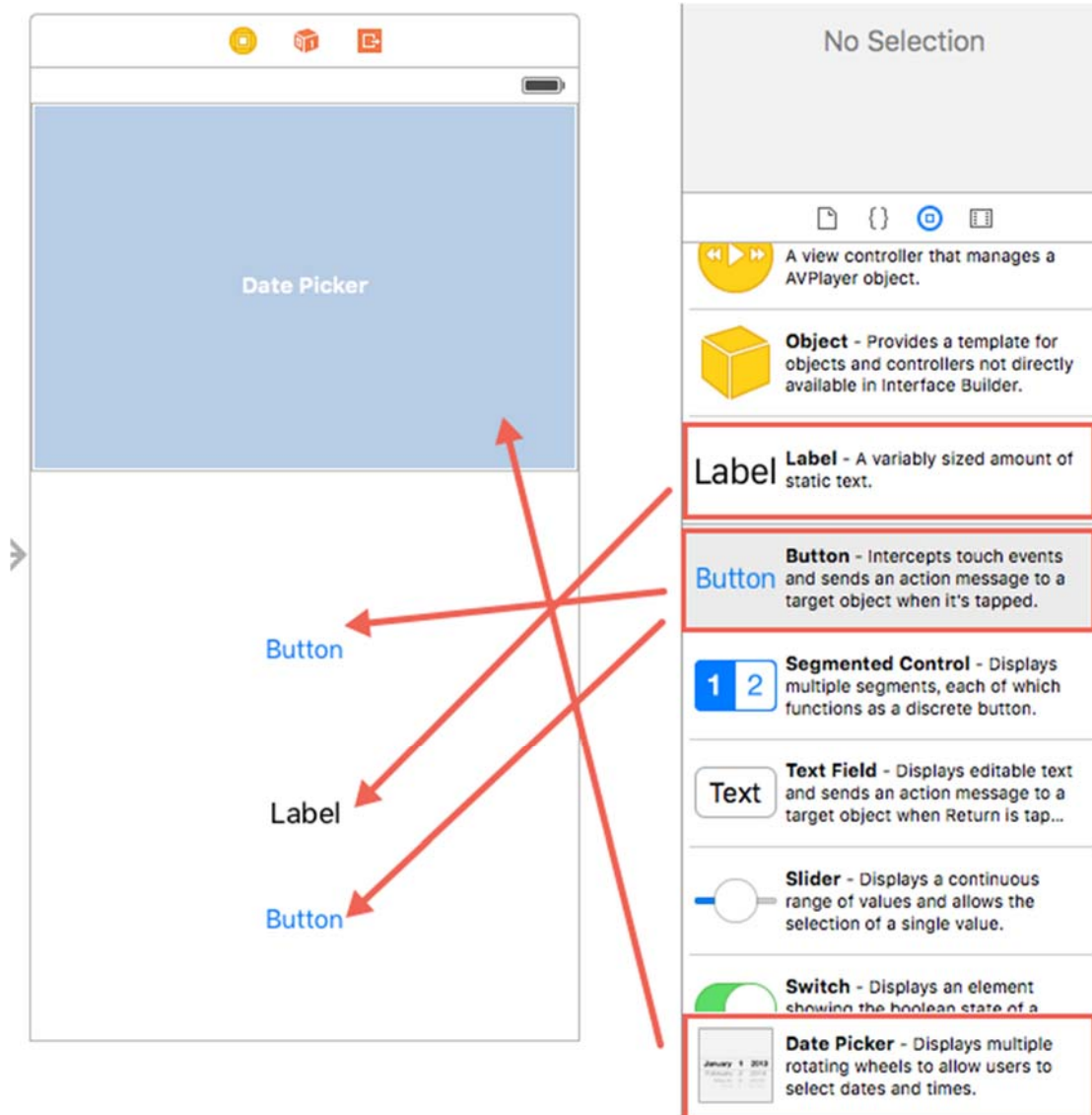
Start

End

6-7

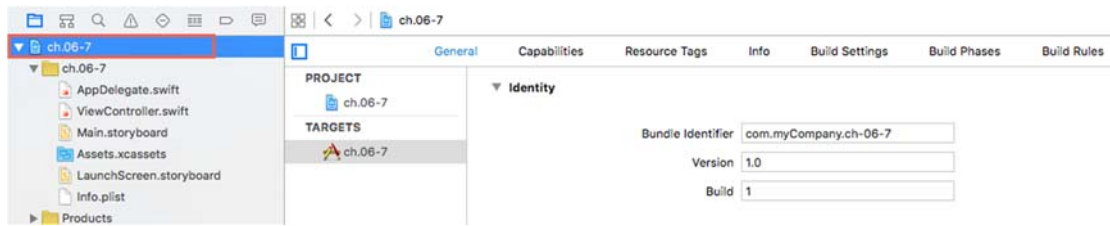
解答

1. 拖曳【Label】、【Button】、【DatePicker】建立在 IB 畫面中，並將 Button 中改為 setAlarm 與 stopAlarm。

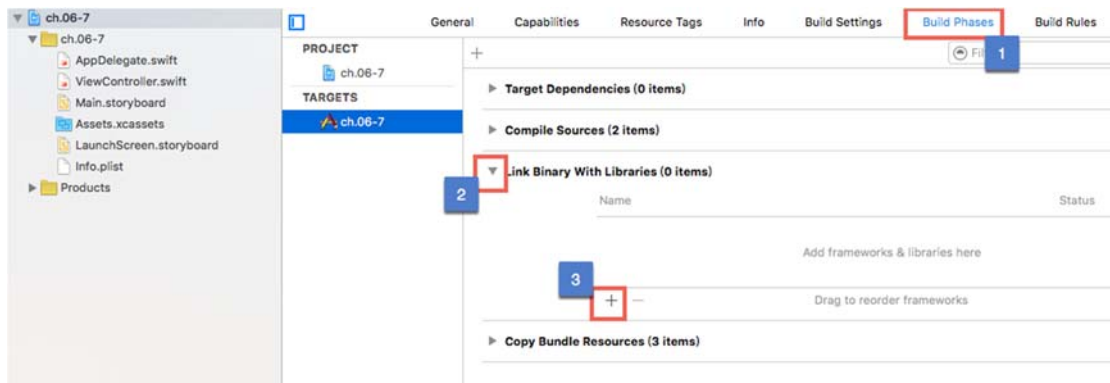


2. 加入音效的函式庫於專案中

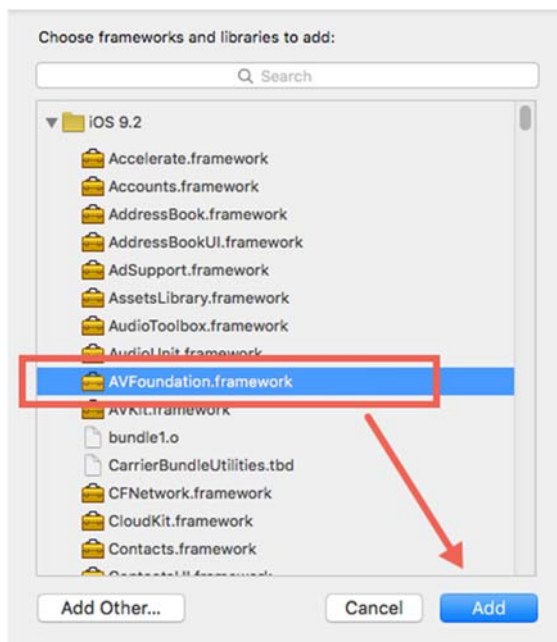
點選左方 專案的最上層，右方會顯示整體專案的相關資訊。

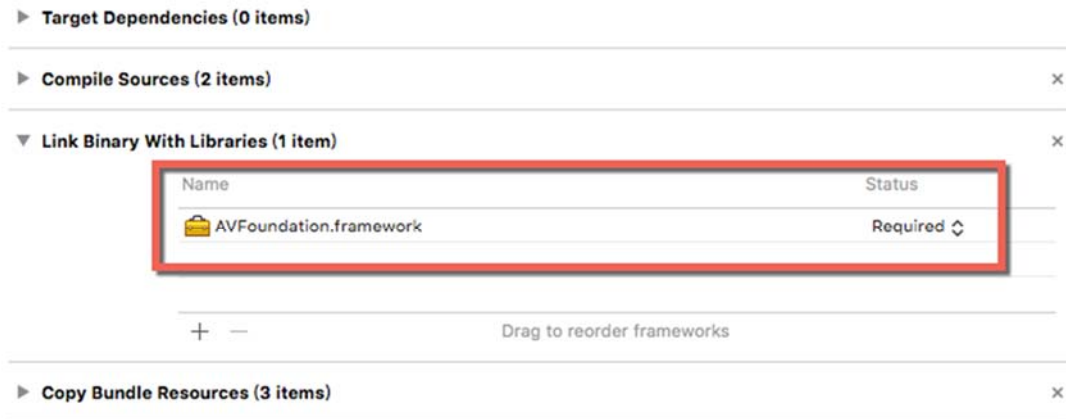


切換至 **Build Phases** 頁籤，並展開 **Link Binary With Libraries**，選取下方的『+』進行新增函式庫。

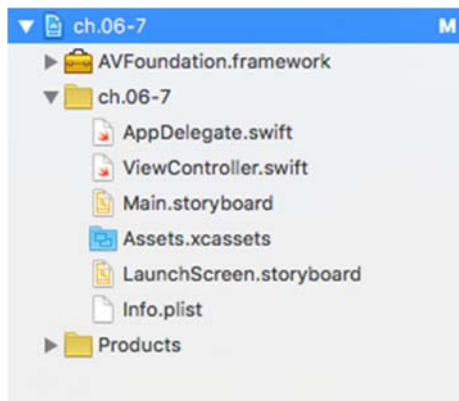


由所列出的函式中選擇 **AVFoundation.framework**，選擇 **Add** 加入。

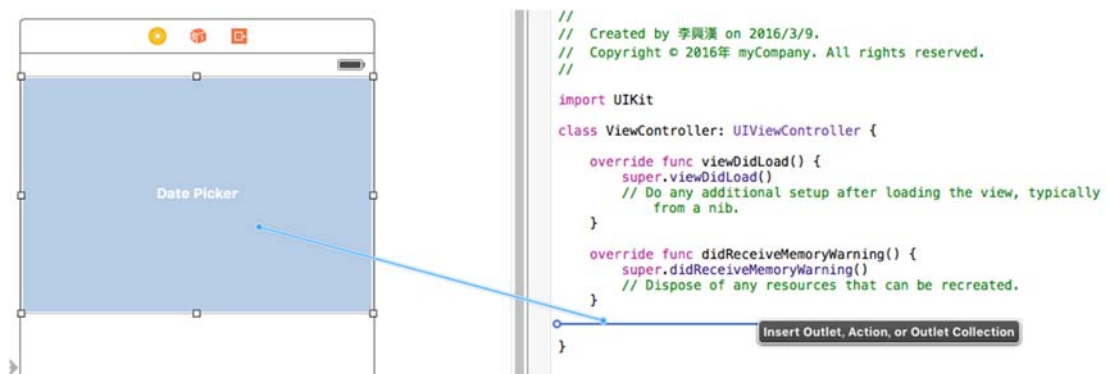




3. 選取【Main.storyboard】



4. 在 IB 視窗上點選【DatePicker】，接著按住【control】鍵，用滑鼠拖曳到右邊視窗與【程式碼】建立連結。

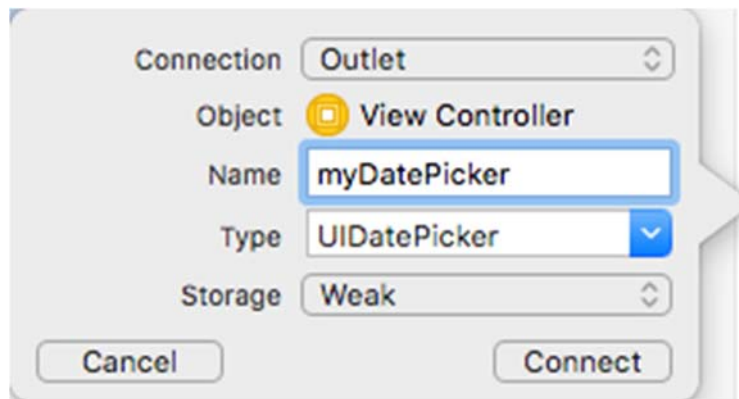


按住【Control】用滑鼠拖曳【DatePicker】元件。

5. 由【物件區】選取之元件，設定名稱如下：

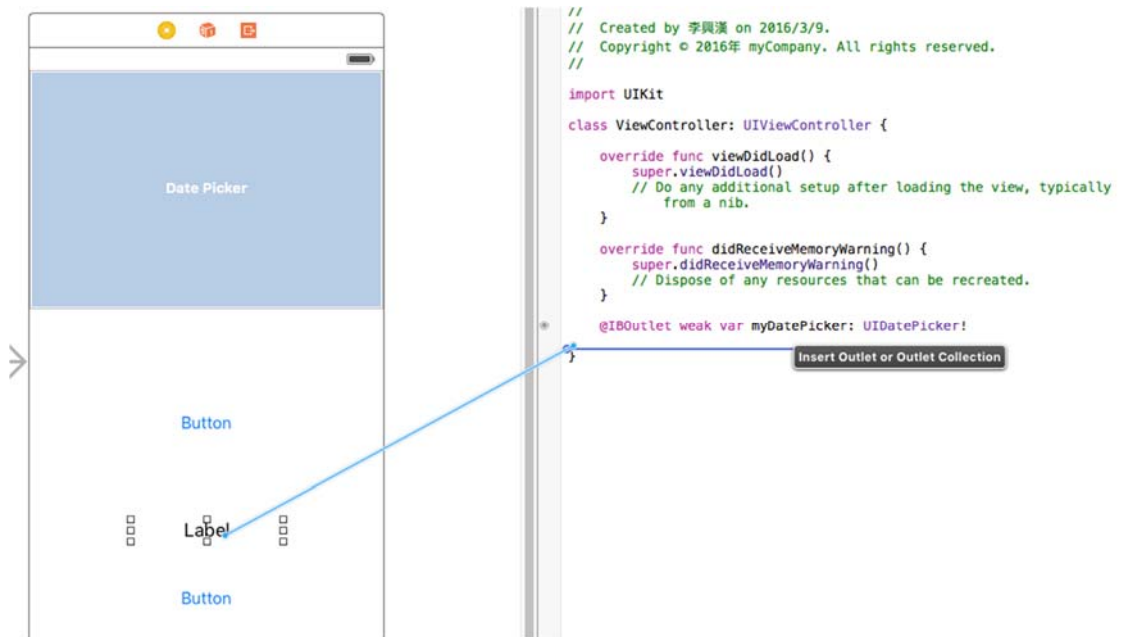
在「Connection」欄位點選【Outlet】、「Name」欄位設定名稱

【myDatePicker】，按【Connect】按鈕。



在拖曳後自動彈跳的視窗

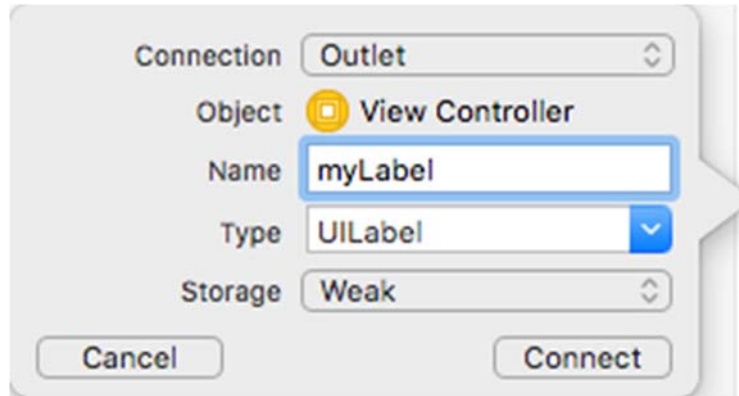
6. 在 IB 視窗上點選【Label】，接著按住【control】鍵，用滑鼠拖曳到右邊視窗與【程式碼】建立連結。



按住【Control】用滑鼠拖曳【Label】元件。

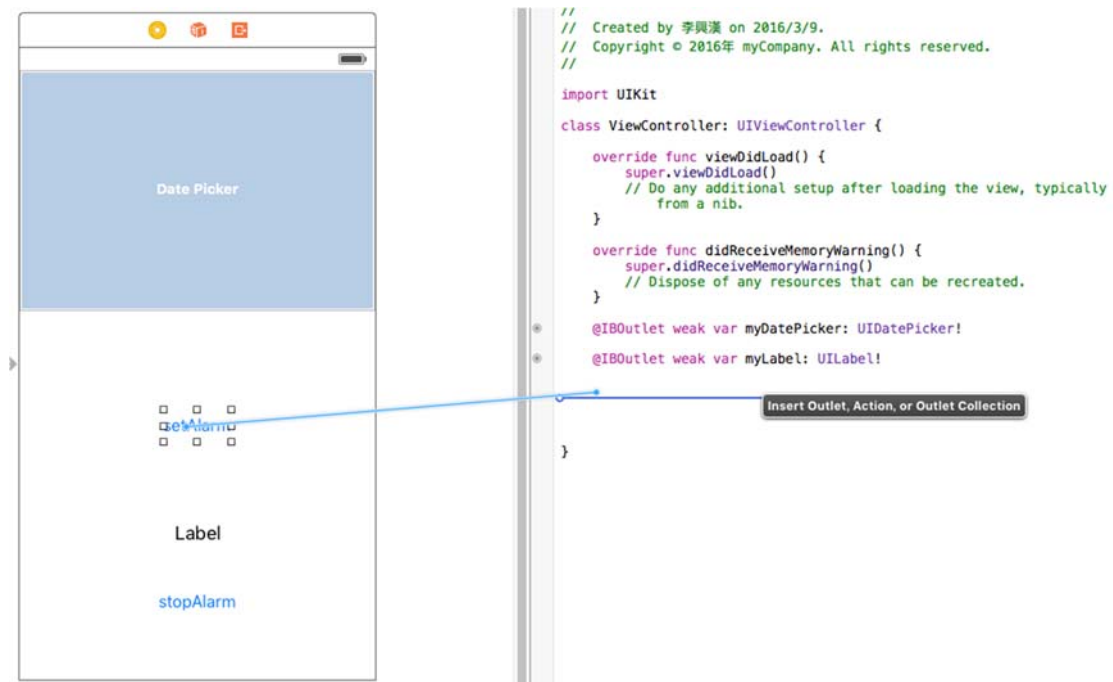
7. 由【物件區】選取之元件，設定名稱如下：

在「Connection」欄位點選【Outlet】、「Name」欄位設定名稱【myLabel】以及在「Type」欄位點選【UILabel】後，按【Connect】按鈕。



在拖曳後自動彈跳的視窗

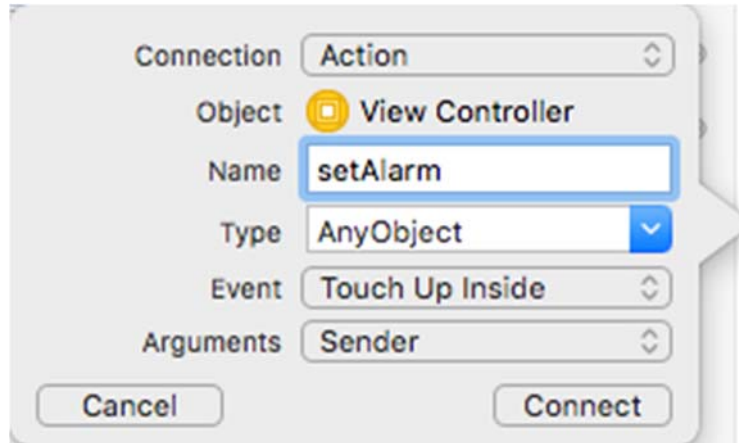
8. 在 IB 視窗上點選 **setAlarm** 【Button】，接著按住 【control】 鍵，用滑鼠拖曳到右邊視窗與 【程式碼】 建立連結。



按住 【Control】 用滑鼠拖曳 **setAlarm** 【Button】 元件。

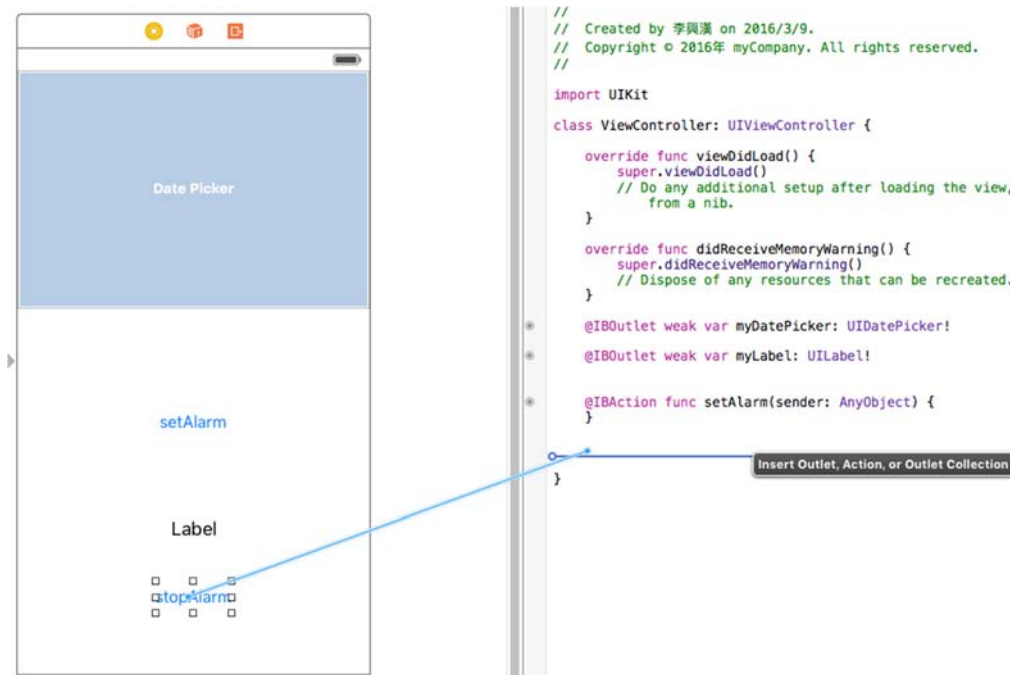
9. 由 【物件區】 選取之元件，設定名稱如下：

在「**Connection**」欄位點選 【**Action**】、「**Name**」欄位設定名稱 【**setAlarm**】，按 【**Connect**】 按鈕。



在拖曳後自動彈跳的視窗

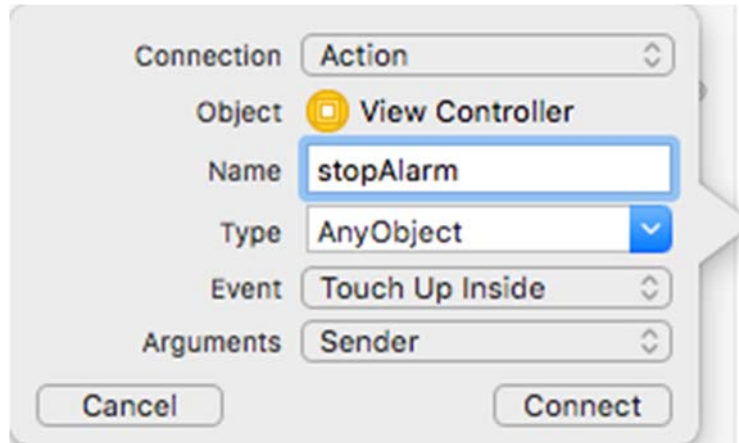
10. 在 IB 視窗上點選 **stopAlarm** 【Button】，接著按住【control】鍵，用滑鼠拖曳到右邊視窗與【程式碼】建立連結。



按住【Control】用滑鼠拖曳 **stopAlarm** 【Button】 元件。

11. 由【物件區】選取之元件，設定名稱如下：

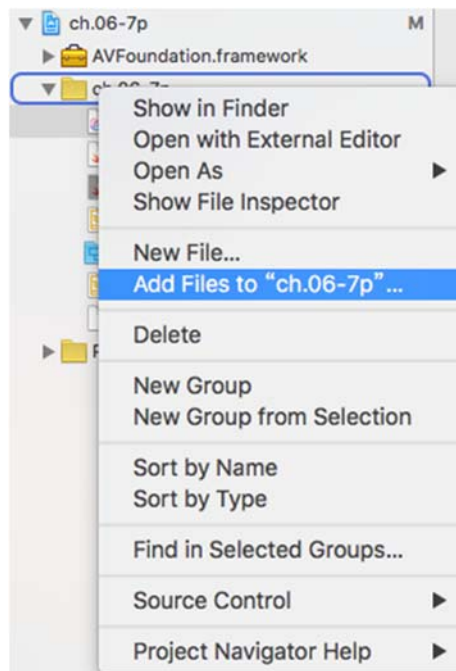
在「Connection」欄位點選【Action】、「Name」欄位設定名稱【stopAlarm】，按【Connect】按鈕。



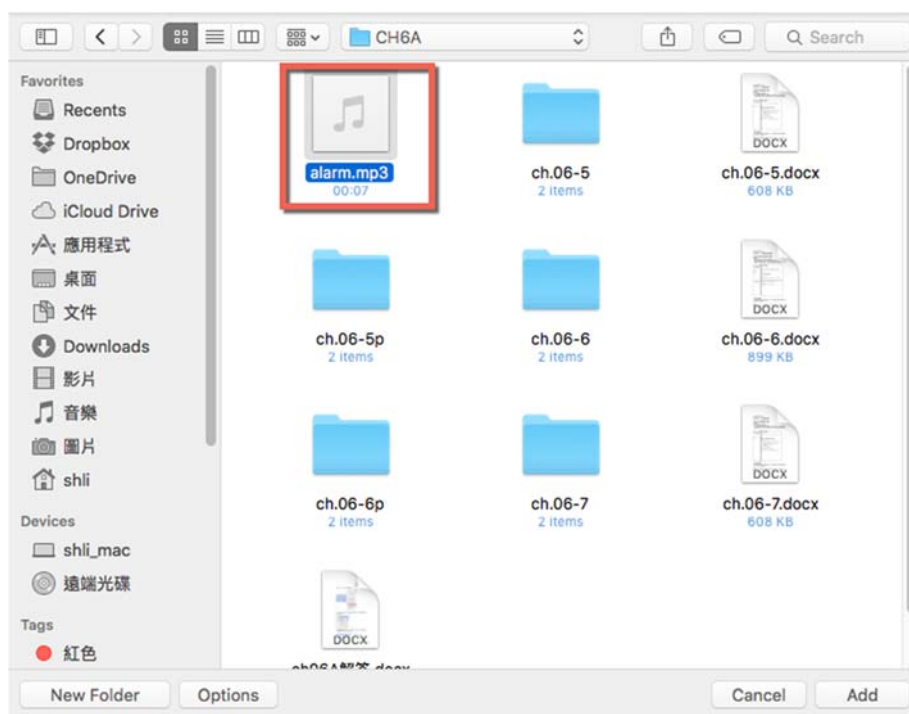
在拖曳後自動彈跳的視窗

12. 加入音效檔 alarm.mp3

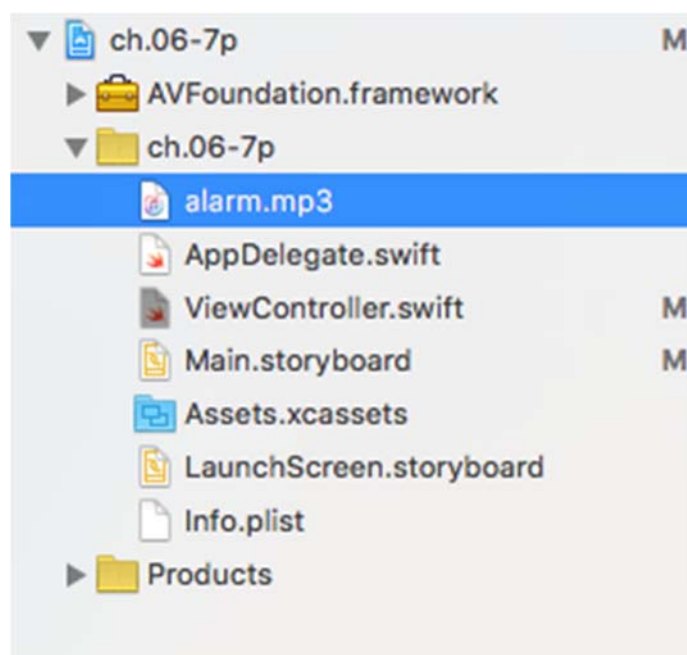
- (1) 在這裡我們選擇一個音效檔，展開 **ch.06-7p** 目錄並且在此目錄中任一位置按右鍵→Add Files to “ch.06-7p...”。



- (2) 在這裡我們選擇選取的檔案，名稱為 <alarm.mp3>，並在選取後按「Add」。



(3) 完成後，接著會看到此 **alarm.mp3**，顯示在 **ch.06-7p** 目錄底下。



(4) 或者，讀者們可以利用拖曳的方式，將音效檔加入到專案內。

將要加入的音效檔案，選取後用拖曳方式加入

13.最後，我們在【紅框】中設定 **swift** 的程式碼在編輯區當中。

手動輸入紅框內的程式

```
import UIKit
import AVFoundation

class ViewController: UIViewController {

    var audioPlayer = AVAudioPlayer()

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.

        NSTimer.scheduledTimerWithTimeInterval(0.05, target: self, selector:
            Selector("checkTime"), userInfo: nil, repeats: true)

        let audioPath = NSBundle.mainBundle().pathForResource("alarm", ofType: "mp3")
        let alarmSound = NSURL(fileURLWithPath: audioPath!)

        do{
            audioPlayer = try AVAudioPlayer(contentsOfURL:alarmSound)
            audioPlayer.prepareToPlay()
            // audioPlayer.play()
        }catch {
            print("Error getting the audio file")
        }
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    @IBOutlet weak var myDatePicker: UIDatePicker!
    @IBOutlet weak var myLabel: UILabel!

    var isAlarmOn = false
    var alarmTime = NSDate()

    @IBAction func setAlarm(sender: AnyObject) {
        let df = NSDateFormatter();
        df.dateFormat = "yyyy/MM/dd HH:mm:ss"
        myLabel.text = df.stringFromDate(myDatePicker.date)

        alarmTime = myDatePicker.date
        isAlarmOn = true
    }

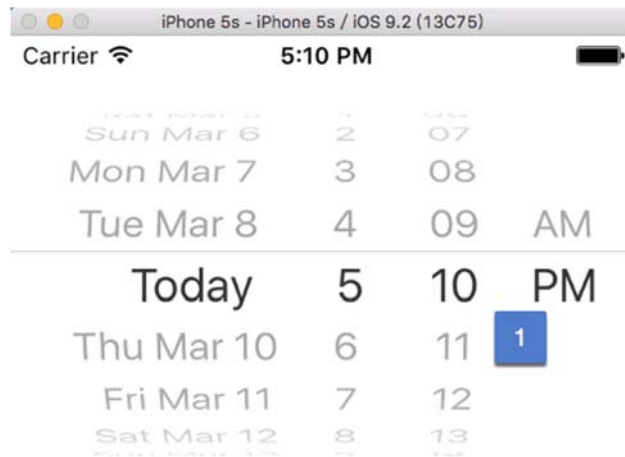
    @IBAction func stopAlarm(sender: AnyObject) {
        isAlarmOn = false
    }

    func checkTime (){
        let now = NSDate()
        if ( alarmTime.timeIntervalSinceReferenceDate <
            now.timeIntervalSinceReferenceDate) && (isAlarmOn == true) {

            audioPlayer.play()

        }
    }
}
```

8. 在啟動「RUN」後，按下 Start 碼錶就開始計時，按下 end 就停止計時。



2
setAlarm

2016/03/09 17:10:21 **3**

stopAlarm **4**