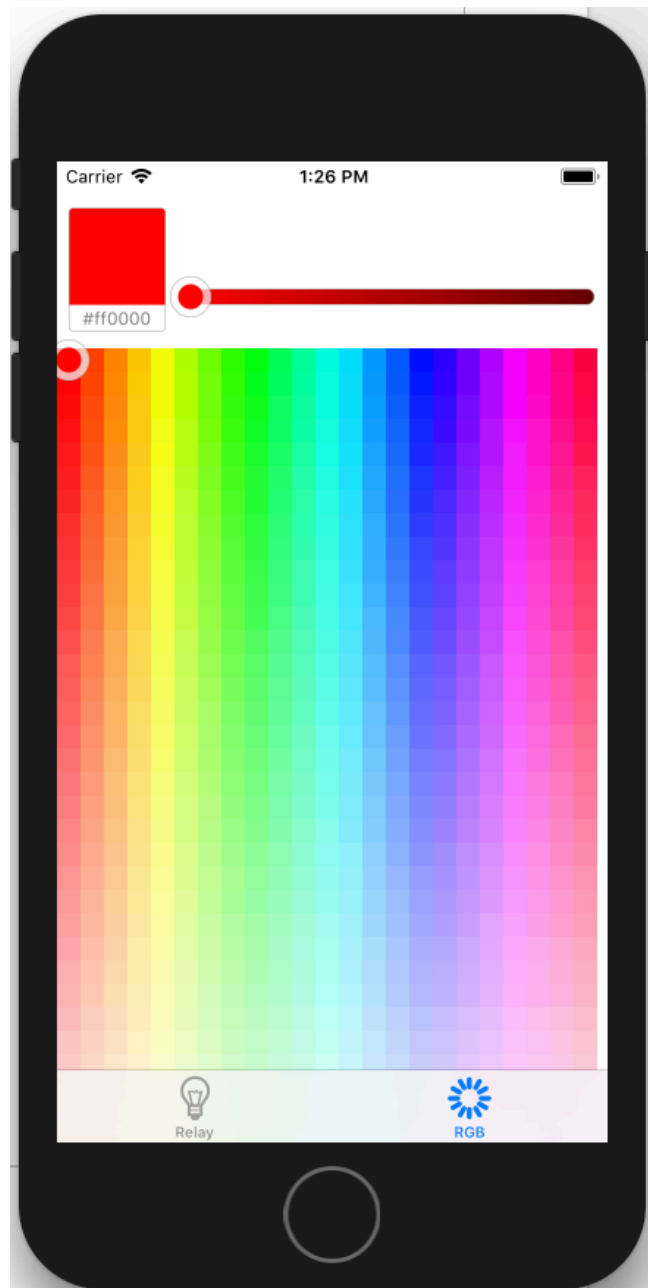


範例arduinoRGB範例說明



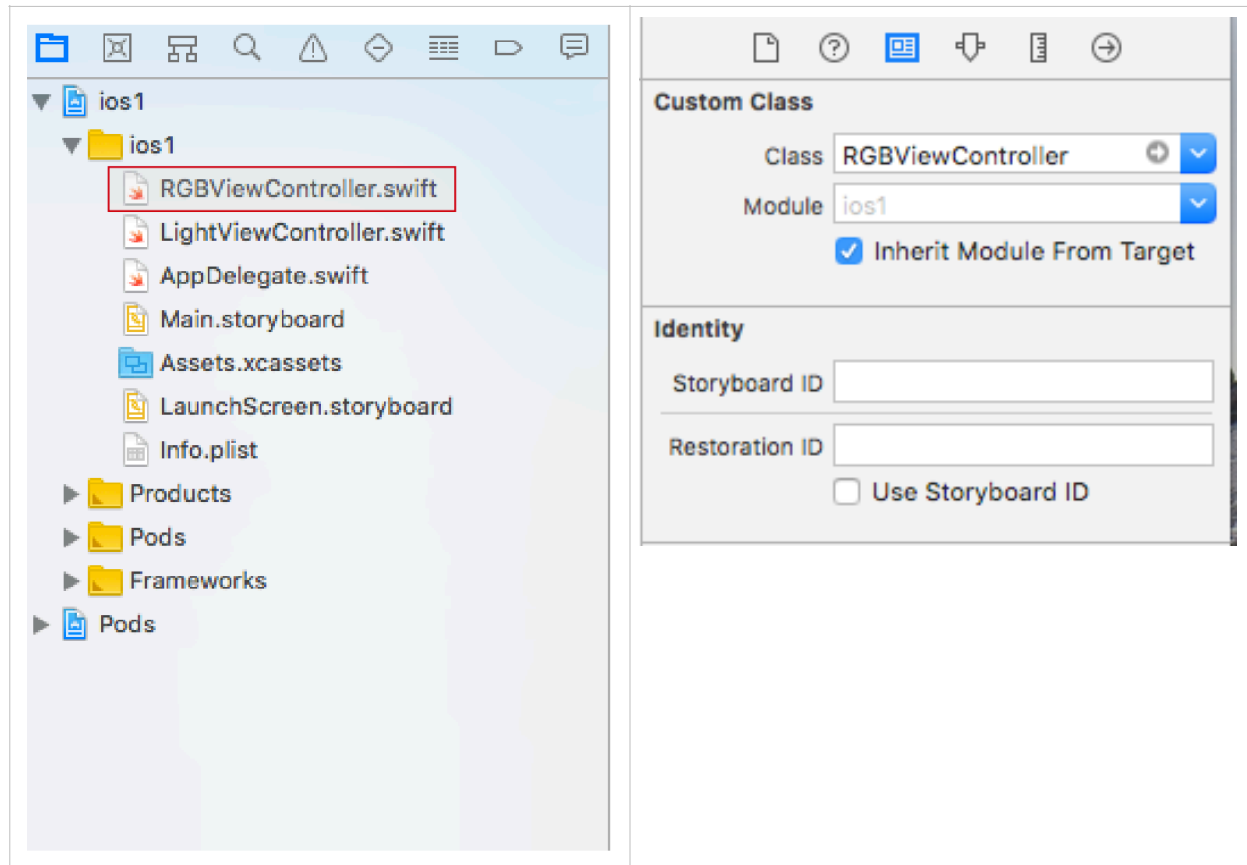
學習目地

- 加入第3方Module
- 連結Firebase資料庫
- 讀取Firebase資料庫節點資料
- 加入監聽器

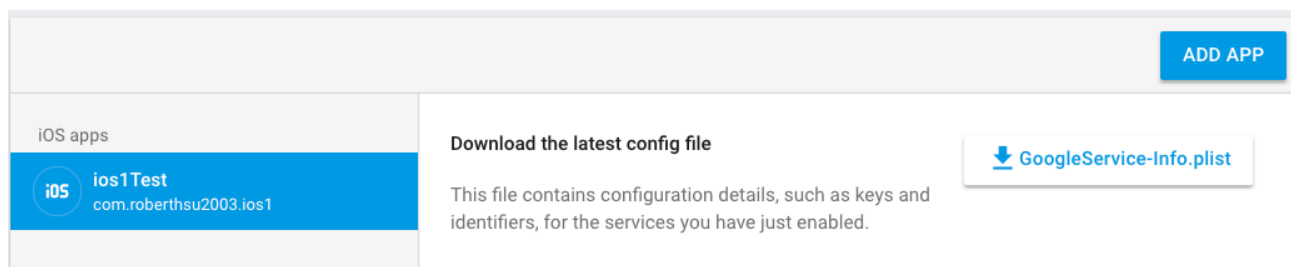
開始準備

解壓縮RGB_start.zip,打開ios1.xcworkspace，解壓縮RGB圖片.zip,將圖譬加入至Assets內。

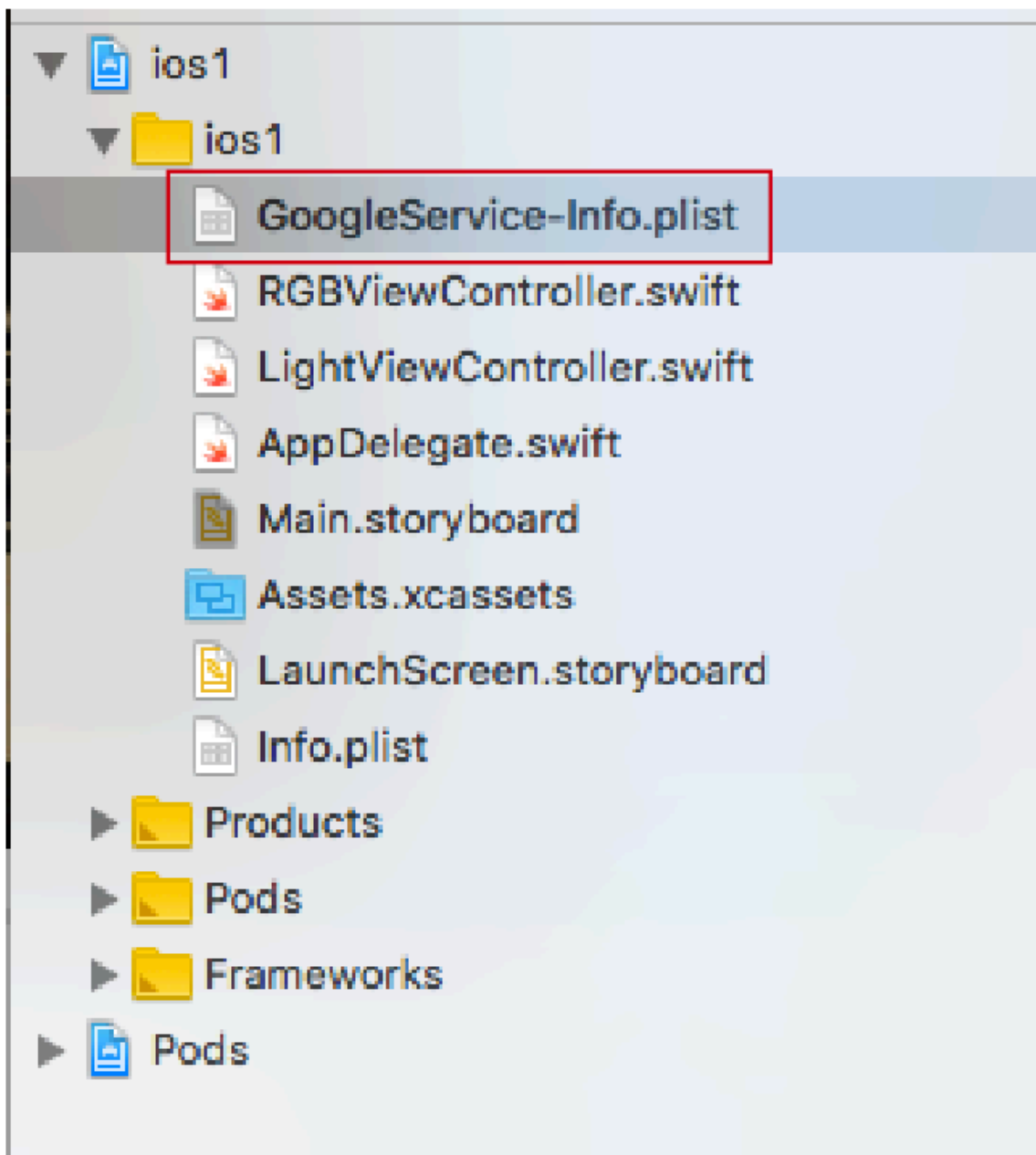
建立RBGViewController，繼承UIViewController，刪除SecondViewController.swift,在Main.storyboard內，將class SeconViewController更改為RGB ViewController。



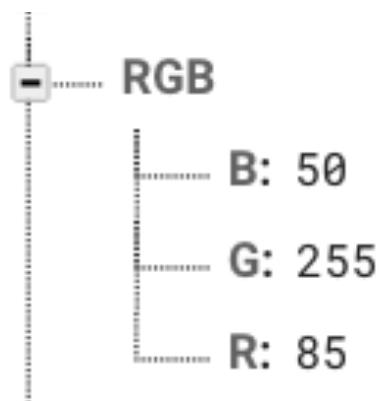
下載Firebase專案內的 GoogleService-info.plist



將GoogleService-info.plist加入至專案內。

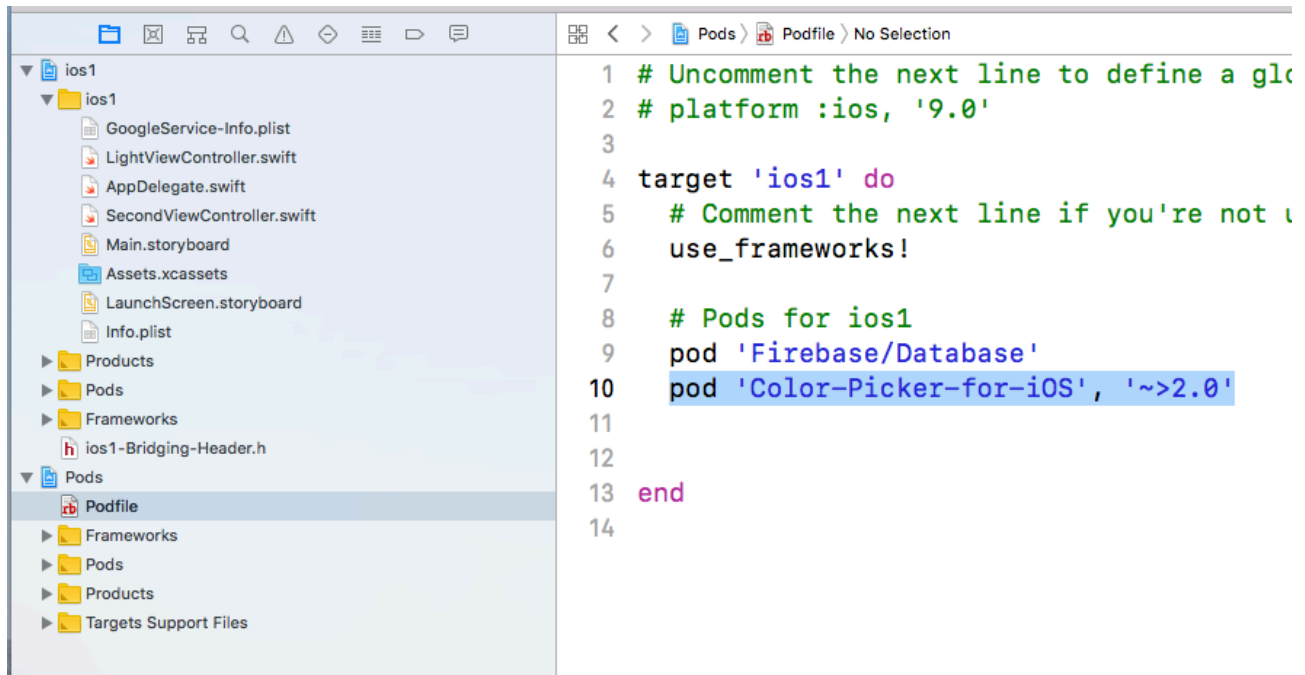


Firebase 設定RGB節點資料



加入第3方Module

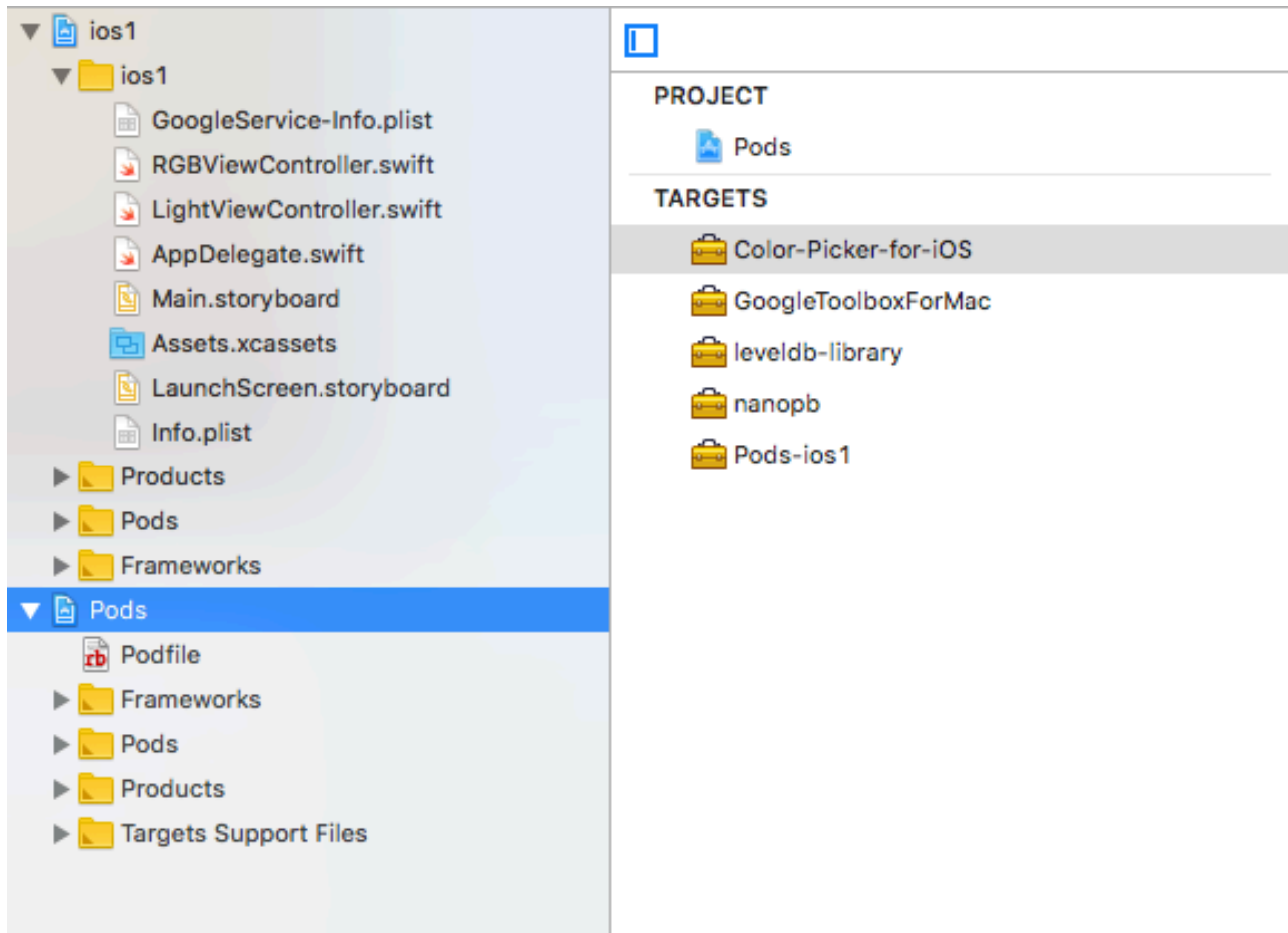
在Podfile內加入pod 'Color-Picker-for-iOS', '~>2.0'，然後關閉專案。



打開終端機軟體，使用cd進入專案資料夾，執行pod install

```
Last login: Wed Oct 25 09:26:14 on console
[xuguotangde-iMac-2:~ roberthsu2003$ cd /Users/roberthsu2003/Google\ 雲端硬碟/iphone2/範例 2_2_arduinoRGB/iphone/ios1_RGB
xuguotangde-iMac-2:ios1_RGB roberthsu2003$ pod install
```

重新打開專案，確認Color-Picker-for-iOS Module有被加入。



打開RGBViewController.swift，並且import Firebase和Color_Picker_for_iOS進入RGBViewController.swift。

```
import UIKit
import Firebase
import Color_Picker_for_iOS
```

在RGBViewController內，建立屬性rgbRef和colorPickerView

```
var rgbRef:DatabaseReference!;
let colorPickerView = HRCOLORPickerView();
```

在ViewDidLoad()內，建立HRCOLORPickerView實體，加入預設顏色，設定大小和x,y座標，並且加入到RGBViewController內的View內，並且加入事件監聽。建立監聽的method colorChange(_:);

```

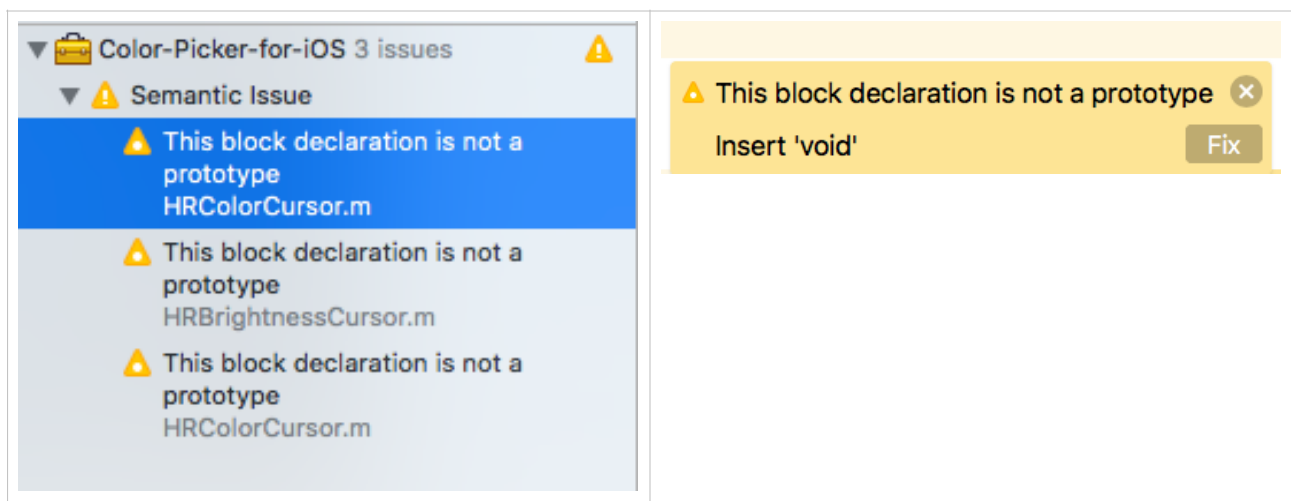
override func viewDidLoad() {
    super.viewDidLoad()
    colorPickerView.color = UIColor.blue;
    colorPickerView.frame = view.frame;
    colorPickerView.frame.origin.y = 20;
    colorPickerView.addTarget(self, action: #selector(colorChange), for:
    UIControlEvents.valueChanged);
    self.view.addSubview(colorPickerView);
}

@objc func colorChange(_ sender:HRColorPickerView){

}

```

執行模擬器，修正警告提示，按Fix



在ViewDidLoad()時間點，建立rgb節點的參考，並且設定一次監聽，取得R,G,B節點的值，並且將值來改變ColorPickerView的顏色。

```

rgbRef = Database.database().reference(withPath: "RGB");
rgbRef.observeSingleEvent(of: .value) { (snapshot:DataSnapshot) in
    let rgbValues = snapshot.value as! [String:Float];
    let r = rgbValues["R"]!;
    let g = rgbValues["G"]!;
    let b = rgbValues["B"]!;
    self.colorPickerView.color = UIColor(red: CGFloat(r/255.0) ,
    green: CGFloat(g/255.0) , blue: CGFloat(b/255.0), alpha: 1);
}

```


使用者，選取顏色時，會執行colorChange(_:)，我們透過HRCOLORPickerView的屬性Color取得顏色，使用getRed(_:green:,blue:,alpha:)取得RGB顏色。參數資料類型為UnsafeMutablePoint<CGFloat>，解決方法為建立var變數的CGFloat，使用CGFloat時，需要在變數前加入&。CGFloat*255運算後一樣是CGFloat，所以必需轉為Int。將值上傳至Firebase console。

```
@objc func colorChange(_ sender:HRCOLORPickerView){
    var rValue:CGFloat = 0.0;
    var gValue:CGFloat = 0.0;
    var bValue:CGFloat = 0.0;
    var aValue:CGFloat = 0.0;
    sender.color.getRed(&rValue, green: &gValue, blue: &bValue, alpha:
    &aValue);
    let r = Int(rValue*255);
    let g = Int(gValue*255);
    let b = Int(bValue*255);

    self.rgbRef.setValue(["R":r,"G":g,"B":b]);
}
```

模擬器執行，並且在Firebase console驗證R:G:B值可否改變顏色，打開ArduinoRGB電源，觀察顏色變化。

