

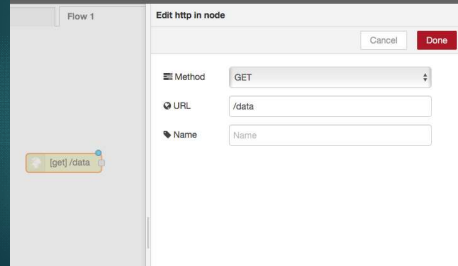
嵌入式系統

Lab 08: 使用 Node-Red Web service 控制 GPIO

銘傳大學 電腦與通訊工程學系 羅嘉寧

建立 Web Server

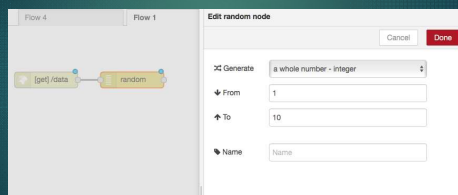
► URL: <http://raspberrypi.local:1880/data>



使用 Node-Red Web service 控制 GPIO

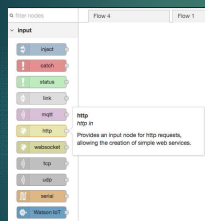
產生回傳值

- 當連到 <http://raspberrypi.local:1880/data> 傳回一 random 值
- Function -> random



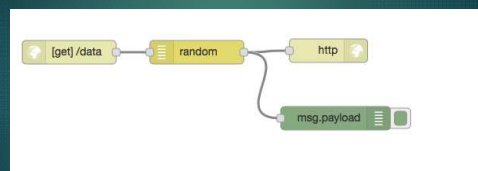
Sample Web Server

- 建一個 Webserver, Input -> http



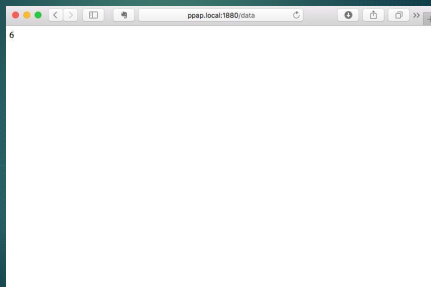
傳值回去

- Output -> http response



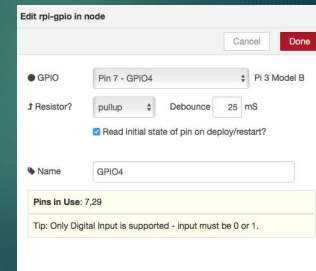
Test

- ▶ URL: <http://raspberrypi.local:1880/data>



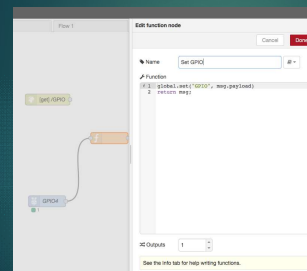
加 GPIO

- ▶ Raspberry_Pi -> rpi gpio



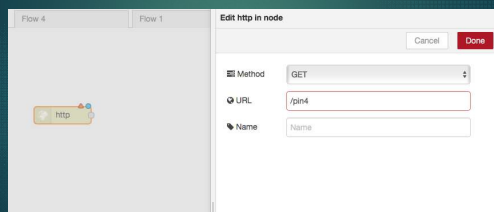
建立 Web Server 傳回 GPIO 狀態

將 GPIO4 存入變數

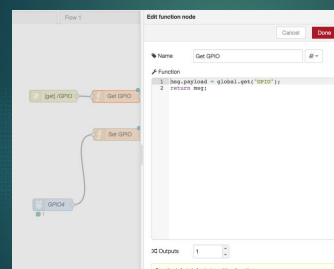


建立 Web Server

- ▶ URL: <http://raspberrypi.local:1880/pin4>

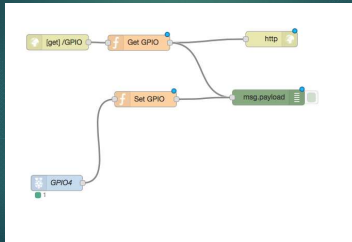


讀取 GPIO 變數



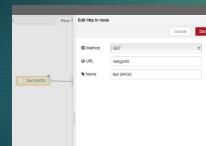
回傳

- ▶ Output -> http response



Start web service

- ▶ URL: <http://raspberrypi.local:1880/setgpio5>

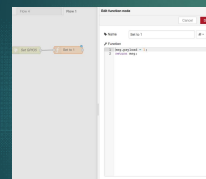


CHECK POINT 1

[HTTP://RASPBERRYPI.LOCA:1880/PIN4](http://raspberrypi.local:1880/PIN4) 可讀取開關狀態

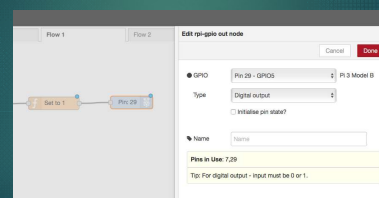
將 GPIO5 設定為 HIGH

- ▶ Function: `msg.payload = 1`



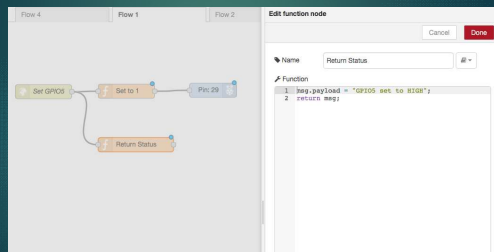
輸出 GPIO5

- ▶ `Gpio_out`

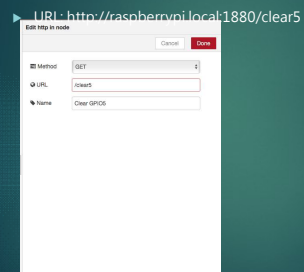


由 Webserver 設定 GPIO

設定回傳值

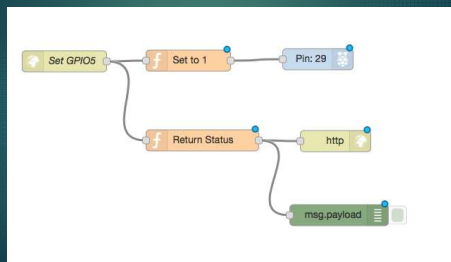


Start another web service



回傳

- ▶ 加入 http response, debug

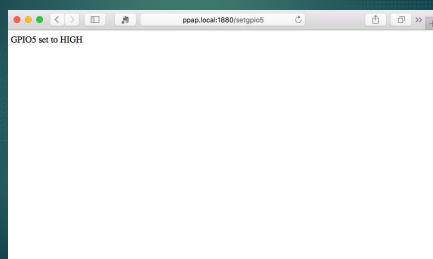


將 GPIO5 設定為 LOW

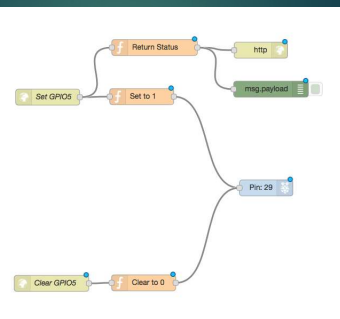


Test

- ▶ <http://raspberrypi.local:1880/setgpio5>

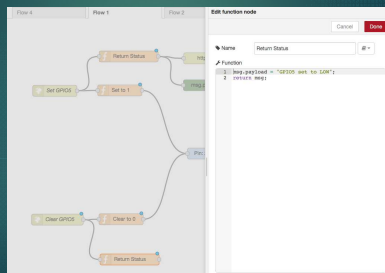


輸出



設定回傳值

▶ 另一個 function



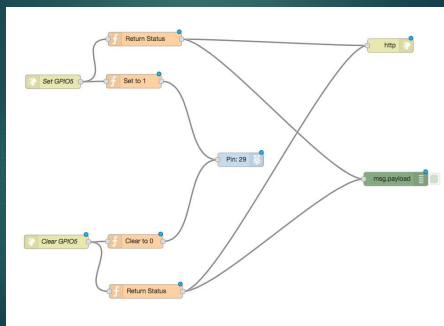
CHECK POINT 2

設定 LED 狀態

[HTTP://RASPBERRYPI.LOCAL:1880/SETGPIO5](http://raspberrypi.local:1880/SETGPIO5)

[HTTP://RASPBERRYPI.LOCAL:1880/CLEAR5](http://raspberrypi.local:1880/CLEAR5)

回傳



Test

▶ <http://raspberrypi.local:1880/clear5>

