

Yapers 嵌入式即時系統開發平台

Yet another platform for embedded real-time systems

指導教授：吳卓俊

成員：王嘉朗 郭明穎 廖鈞榆



國立屏東大學
National Pingtung University



資訊工程學系
Dept. of Computer Science
& Information Engineering



RESL@NPTU
即時與嵌入式系統實驗室
Real-Time & Embedded Systems Lab.



嵌入式系統開發的挑戰與回應



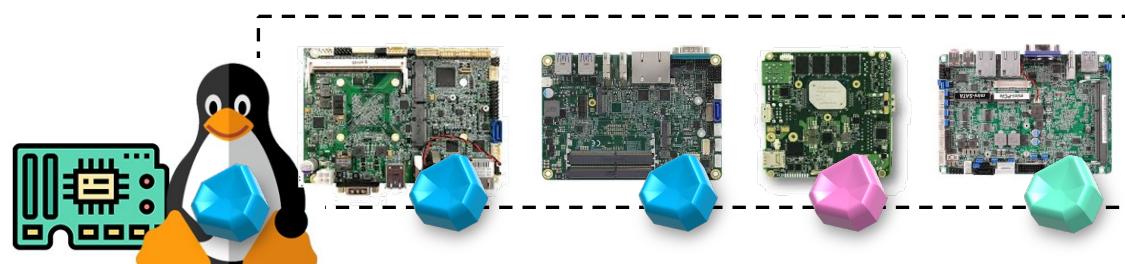
硬體多樣性

同樣的功能在不同硬體環境下，需要開發成不同的程式碼



嵌入式作業系統(Embedded OS)

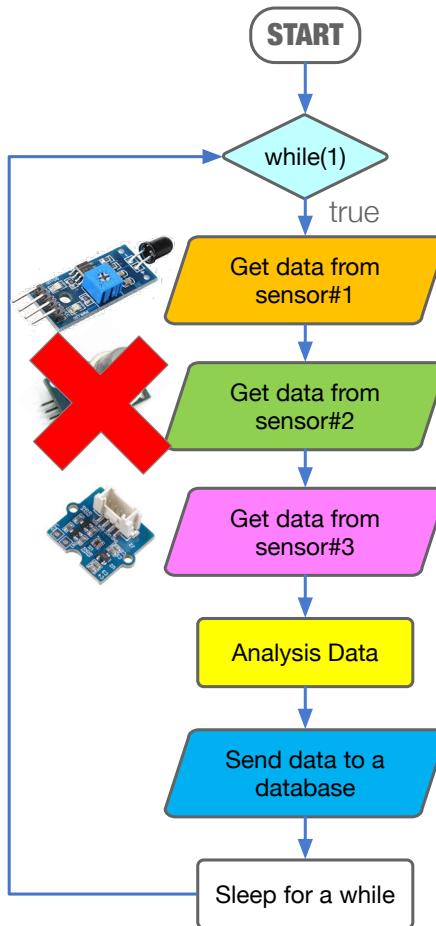
同樣的功能可以在支援相同作業系統的硬體上執行，
但仍然需要針對不同的硬體環境修改程式



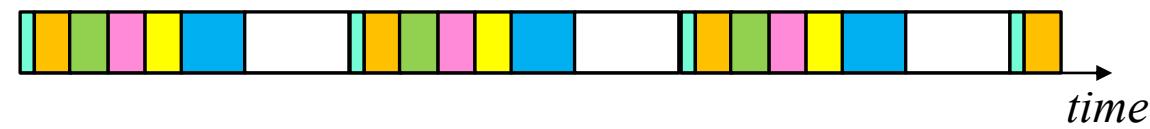
Embedded Linux



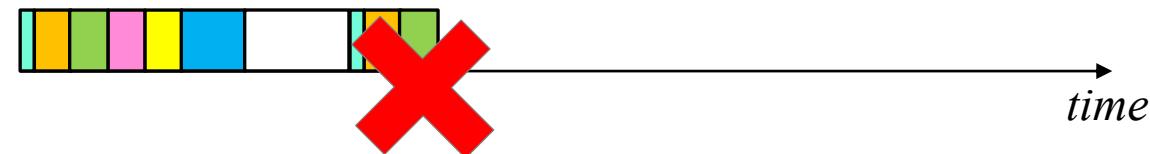
嵌入式軟體設計的挑戰與回應



- 使用單一迴圈反複執行相關功能

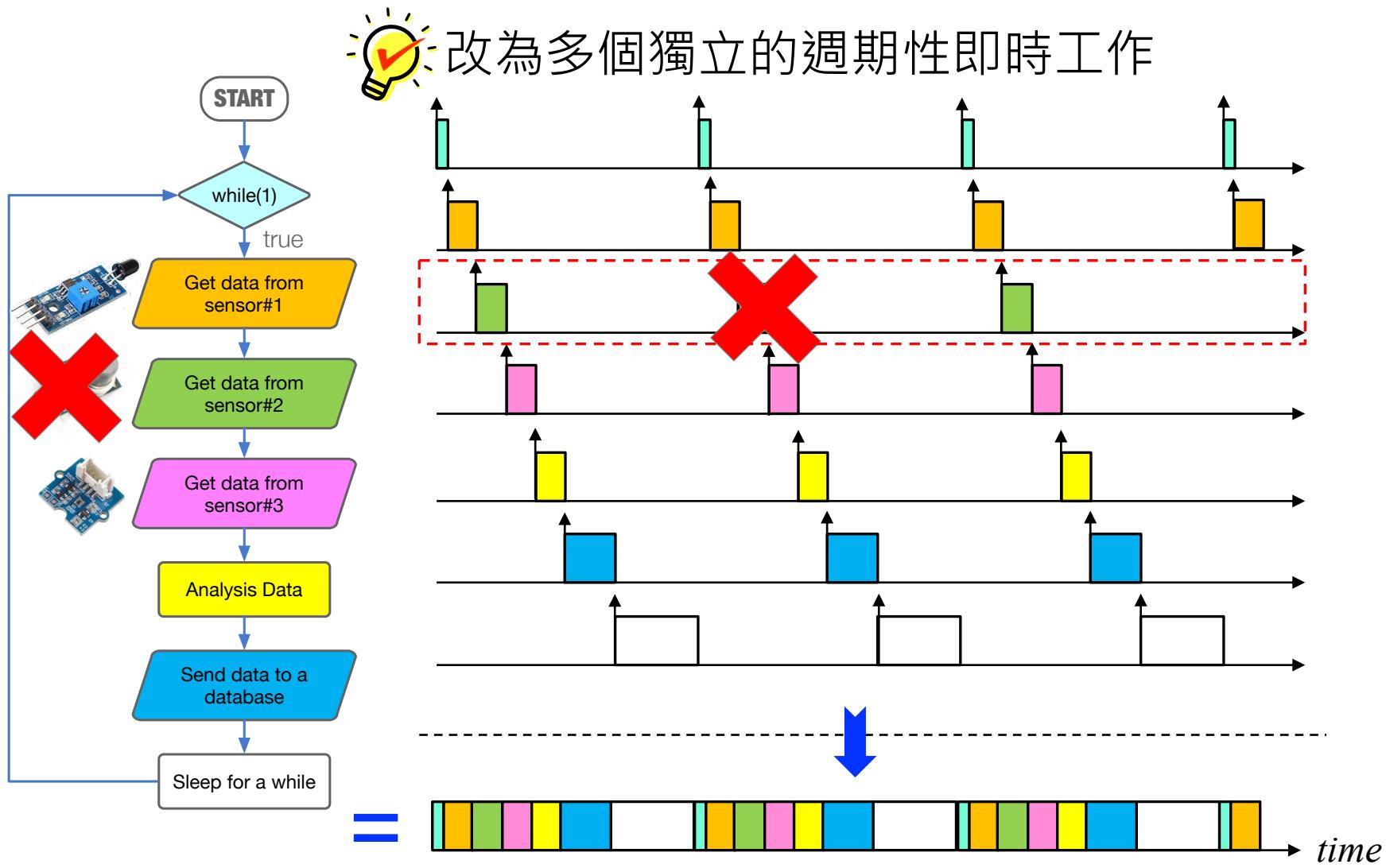


發生單一錯誤時，
將導致所有程式功能都停止執行！





嵌入式軟體設計的挑戰與回應

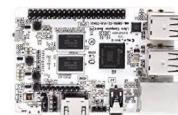




Yapers 嵌入式即時系統平台



Raspberry Pi



Amlogic SBCs



Tinker Board



Udoobolt

可運行Linux的嵌入式開發板
(目前支援Raspberry Pi 4B)

Yet another Platform for Embedded Real-time Systems

嵌入式軟體的開發平台與運行平台

系統組態
工作定義
工作開發

C/Python API
Shared Variables
Shared Files

週期性即時工作

事件驅動的即時工作

由多個工作所組成的軟體系統

SafeNPTU
ecoScout

Earliest Deadline First (EDF)
Rate Monotonic (RM)

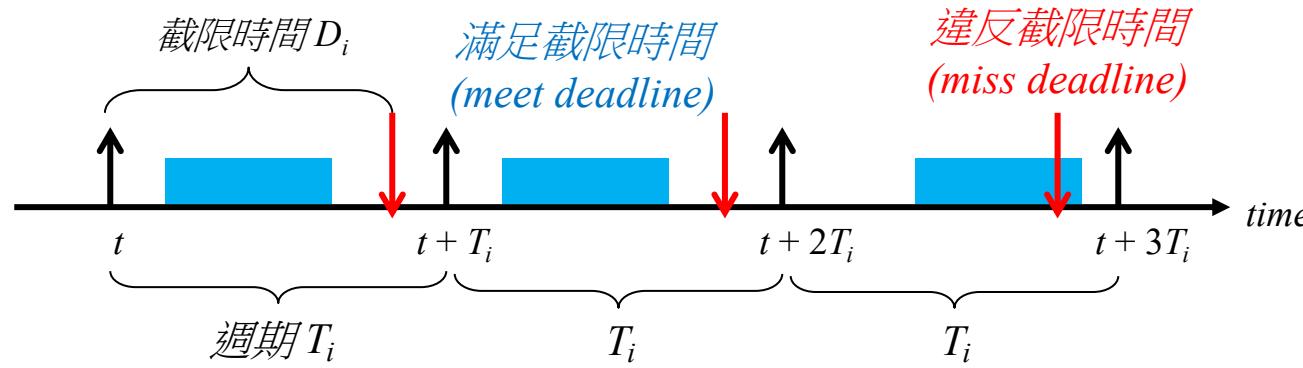
工作排程與運行

Yapers Database
Yapers Server
Yapers Web



即時系統(Real-Time Systems)

- 考慮具有時間限制的工作，除正確完成工作執行外，亦追求在特定時間內完成工作的執行
- 週期性即時工作模型* – 即時工作 τ_i 具有以下參數：
 - T_i ：週期
 - D_i ：截限時間
 - C_i ：最差情況下的執行時間

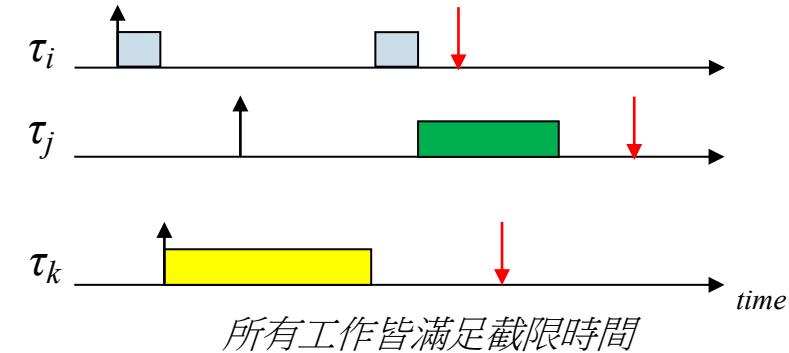
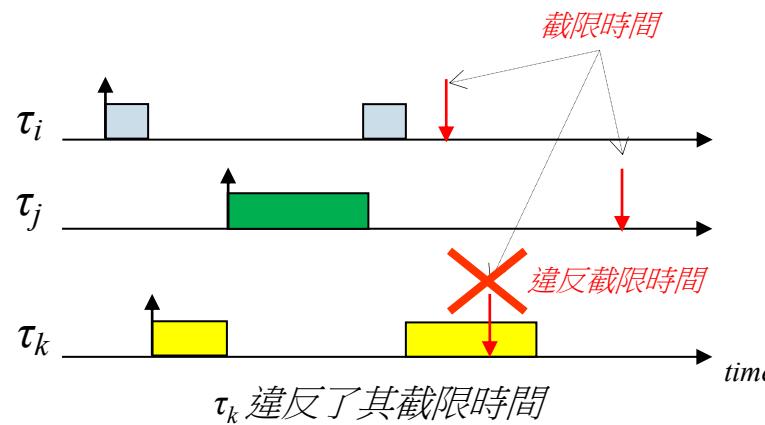


* C. L. Liu and James W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment," Journal of the ACM, Vol. 20, Issue 1, pp. 46-61, Jan. 1973.



即時工作排程

- 對一組週期性即時工作 $T = \{\tau_1, \tau_2, \dots, \tau_n\}$ ，**即時工作排程方法**的作用就是決定工作執行的順序並保證所有工作都能滿足時間限制

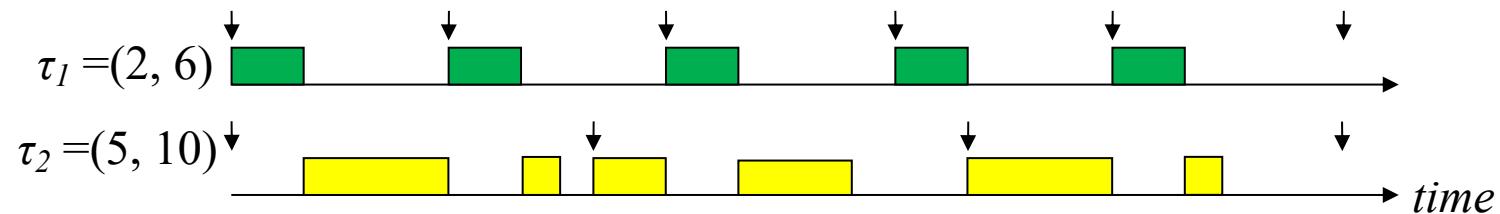




著名的即時工作排程方法

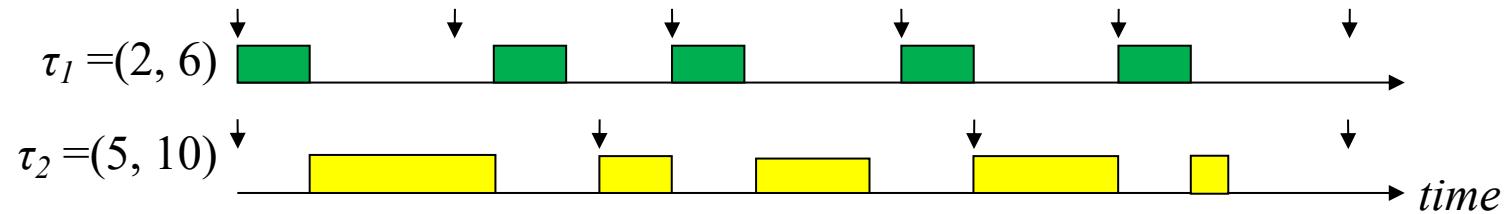
■ 固定優先權的Rate Monotonic (RM)方法

- Offline方法 — 在系統開始運作前即可決定優先權
- 依據工作週期的長短給定優先權 — 週期愈短者其優先權愈高



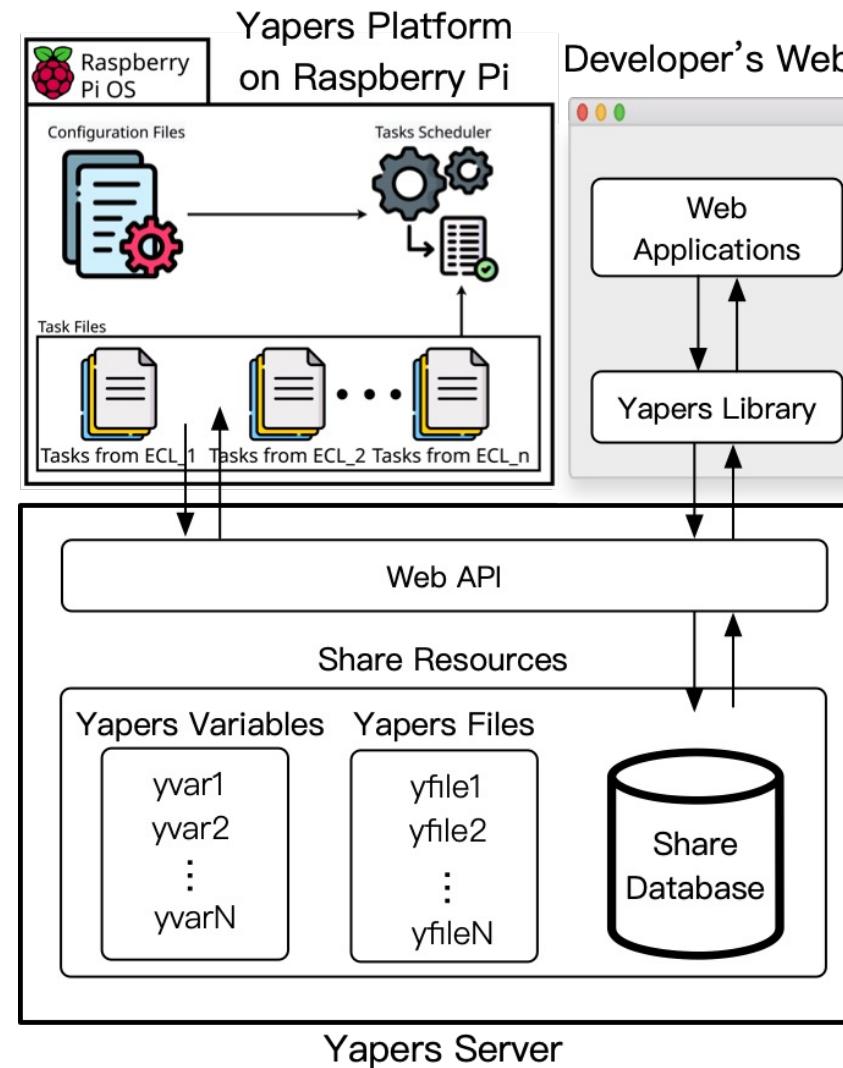
■ 動態優先權的Earliest Deadline First (EDF)方法

- Online方法 — 在系統運行時才決定優先權
- 在系統運行時，依據當時每個在等待佇列中的工作距其截限時間的遠近決定優先權 — 距截限時間愈近者其優先權愈高



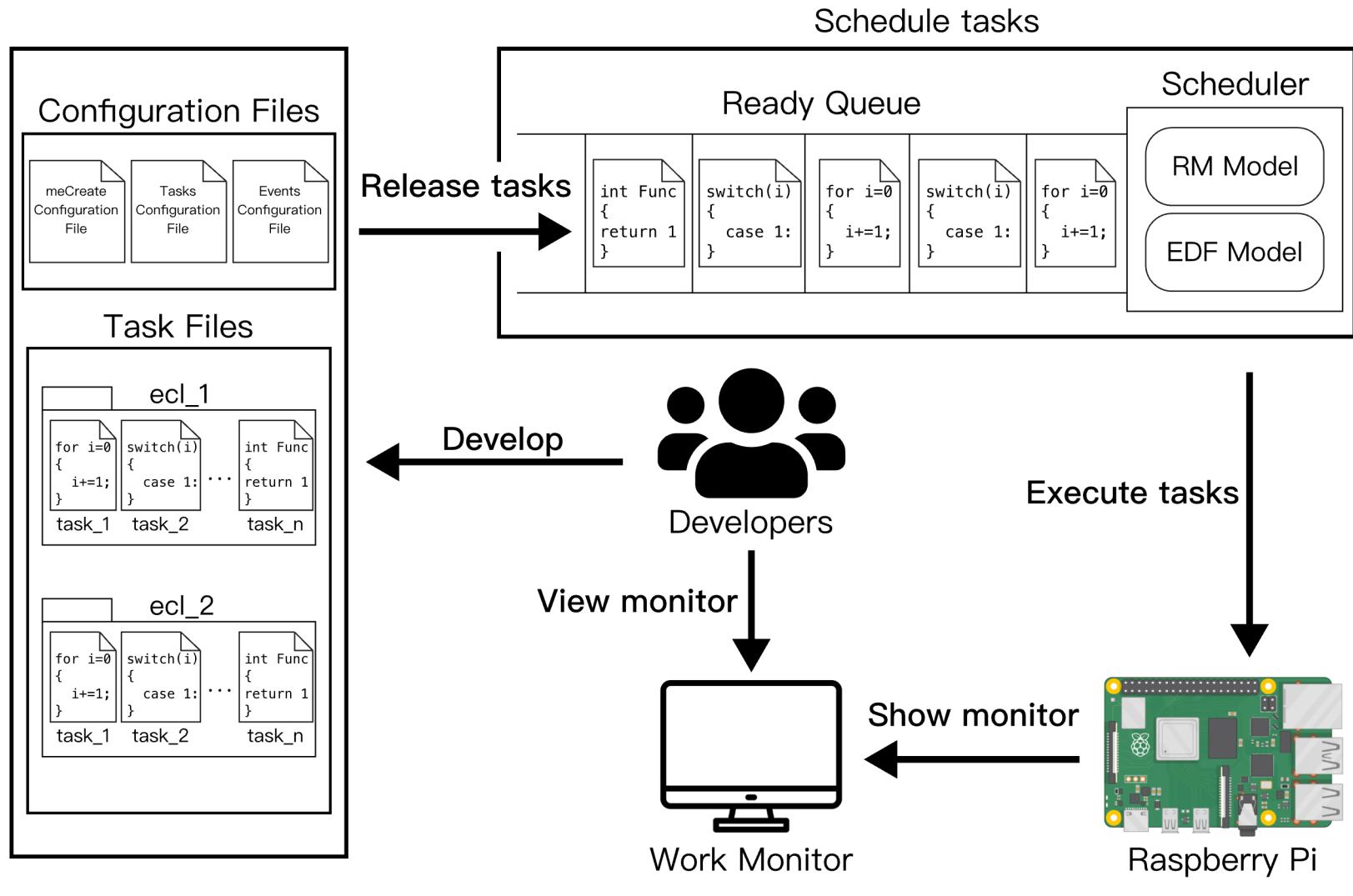


Yapers 平台架構





Yapers 開發流程





節能層級與工作版本

- 每個工作都必須有兩種運行版本
- 以 RE 代表系統剩餘的電池電量

執行較**複雜但高精確度**的工作版本， $if RE \geq 33\%$



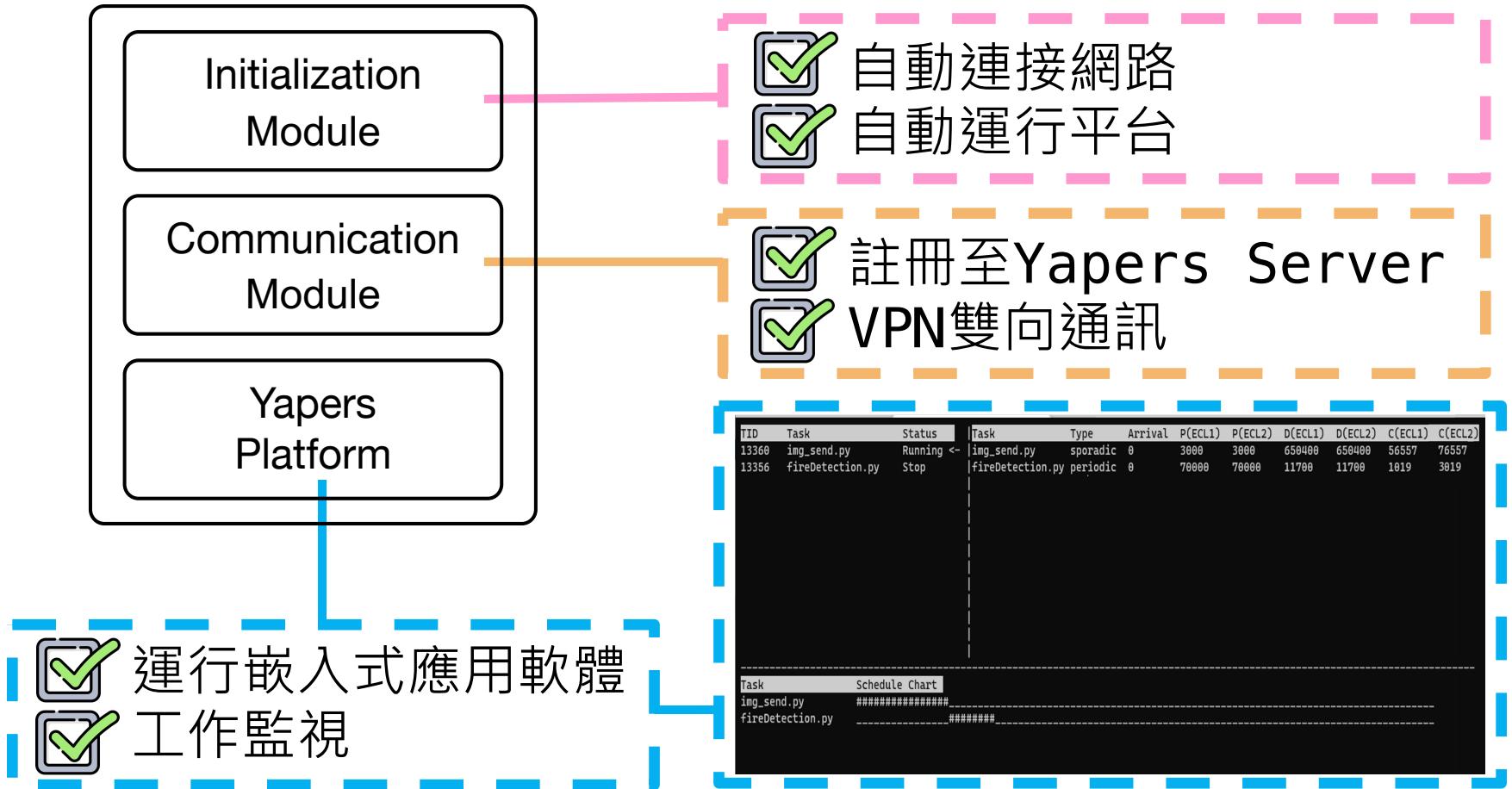
執行較**簡單但低精確度**的工作版本， $if RE < 33\%$





嵌入式作業系統端的修改

Raspberry Pi OS

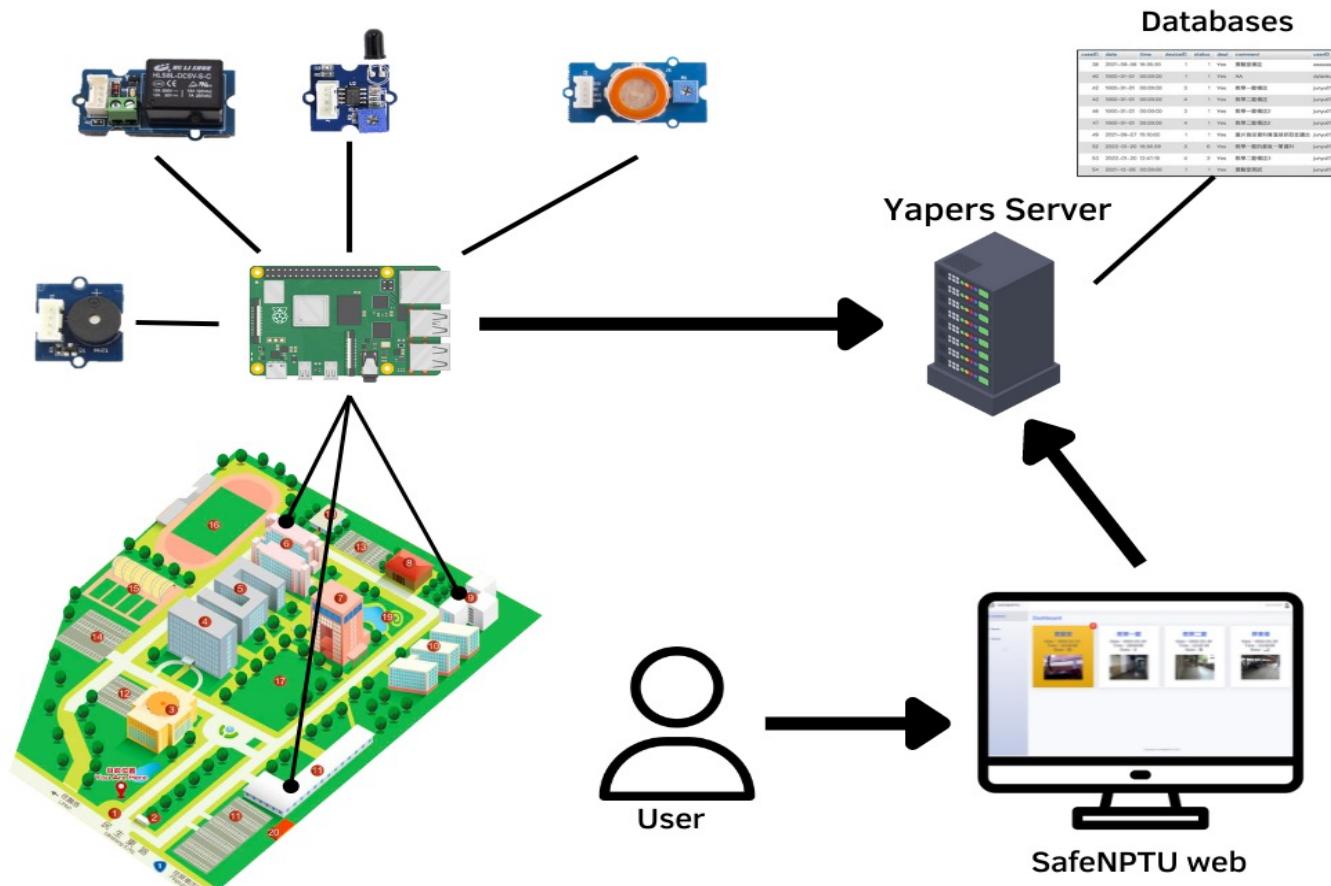


Yapers



Example#1: SafeNPTU

一個用於校園安全的嵌入式應用系統





工作清單

名稱	類型	說明
fire_detect.py	periodic	檢查火焰感測器狀態
take_photo.py	periodic	定期拍攝照片 並回傳伺服器
alerts.py	sporadic	發出警報聲
build_case.py	sporadic	在資料庫中建立案例 並儲存現場照片
get_audio.py	sporadic	從伺服器取得聲音
play_audio.py	sporadic	播放聲音檔案

... and more!





使用Python API開發工作



fire_detect.py

```
# Import modules
from yaperslib import gpio
from yaperslib import device_resources as dr

# Define variables
pin = 5
setmode = 'BCM'
pull = 'DOWN'
yvar_num = 1

# Task description
try:
    isNoFire = gpio.input(pin, setmode, pull)
    if not isNoFire:
        dr.set_device_yvar(yvar_num, 1, 'isFire')
except:
    gpio.cleanup()
    raise
```



使用Python API開發工作



fire_detect.py

```
# Import modules
from yaperslib import gpio
from yaperslib import device_resources as dr

# Define variables
pin = 5
setmode = 'BCM'
pull = 'DOWN'
yvar_num = 1

# Task description
try:
    isNoFire = gpio.input(pin, setmode, pull)
    if not isNoFire:
        dr.set_device_yvar(yvar_num, 1, 'isFire')
except:
    gpio.cleanup()
    raise
```

- dr.set_device_yfile
- dr.set_device_yvar
- dr.get_device_yvar
- gpio.input
- gpio.output
- gpio.cleanup
- event
- ...



工作定義檔

工作名稱	工作類型	抵達時間	週期	截限時間	最差執行時間
fire_detect.py	periodic	0	3000	3000	170
gas_detect.py	periodic	0	3500	3500	210
take_image.py	periodic	0	5000	5000	1950
alerts.py	sporadic	0	100	100	1160
build_case.py	sporadic	0	101	101	2850
get_audio.py	sporadic	0	102	102	4000
play_audio.py	sporadic	0	103	103	30000
sl_detect.py	periodic	0	9000	9000	650
lamp_on.py	periodic	0	5000	5000	740
...					

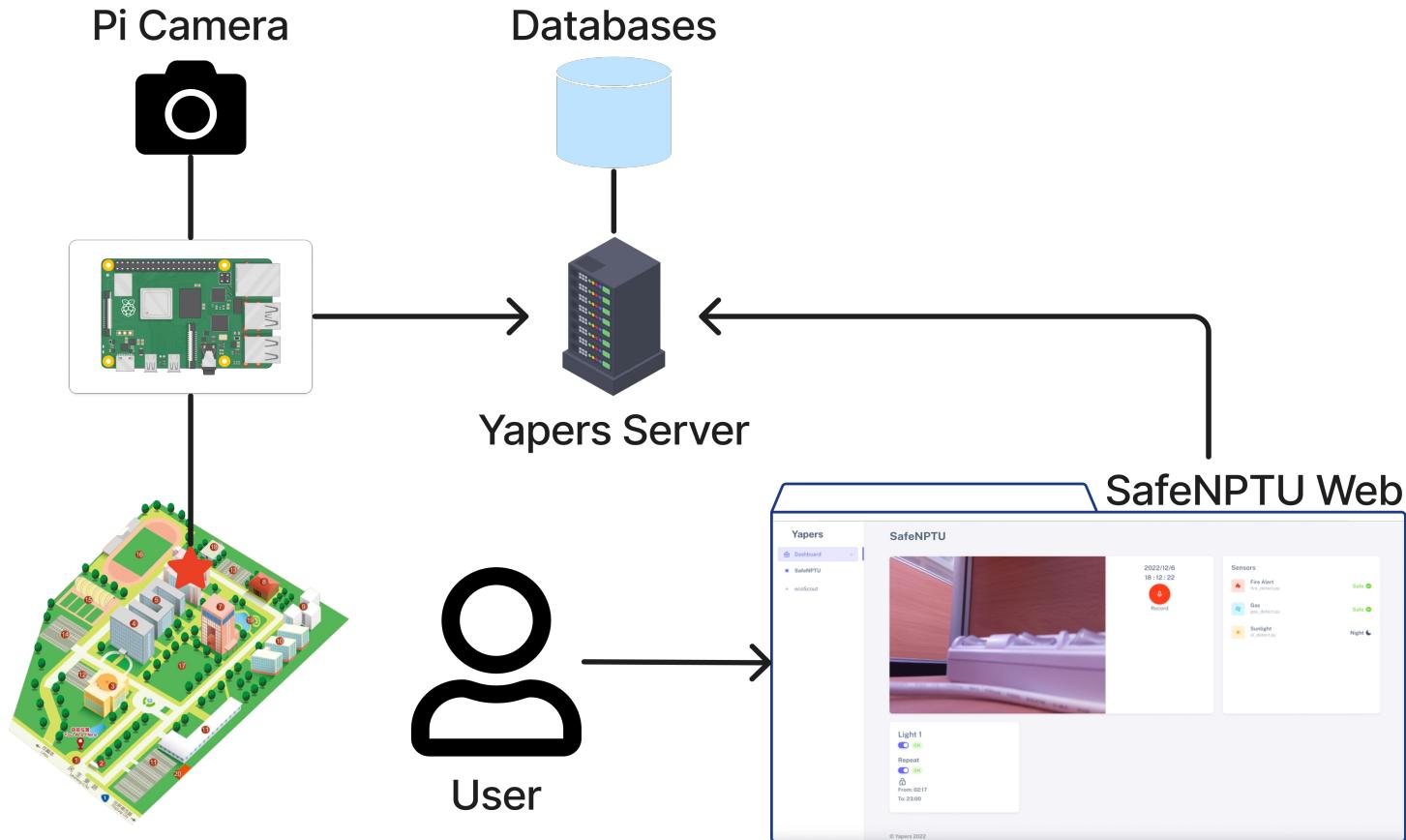


組態檔

```
# 工作是否可被搶佔  
isPreemptive = yes  
  
# 工作的排程方法，支援RM和EDF  
scheduling_algorithm = RM  
  
# 節能層級的數量  
# 當定義為N時，層級編號為ECL_1, ECL_2, ..., ECL_N  
ecl_number = 2  
  
# 切換節能層級的範圍  
ecl_1 = [0,33]  
ecl_2 = (33,100]
```



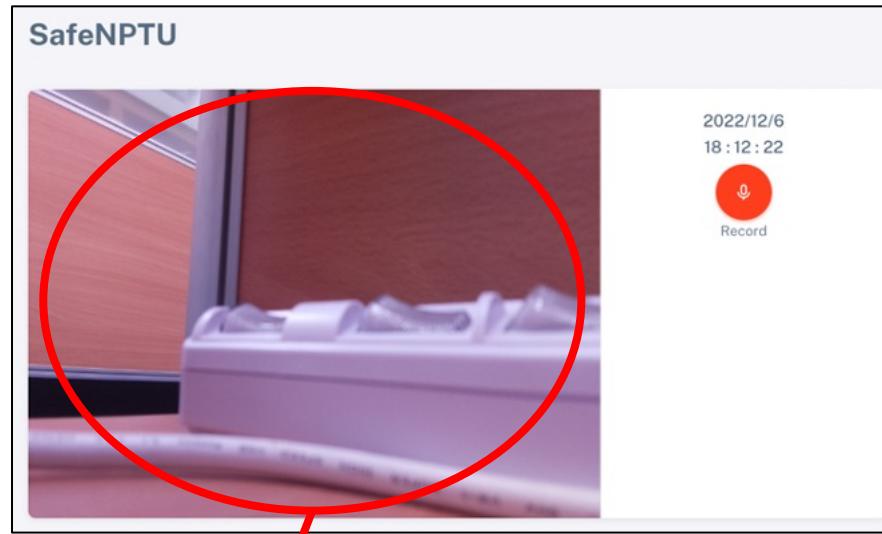
SafeNPTU Web



Yapers



Yapers Web API



```
let changeimg = function(did, yfile_num){  
    get_server_file(did, yfile_num).then((yfile) => {  
        document.getElementById('img').src =  
            `data:image/png;base64, ${yfile}`;  
    });  
};  
setInterval(changeimg, 5000, 4, 1);
```



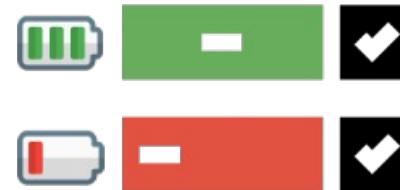
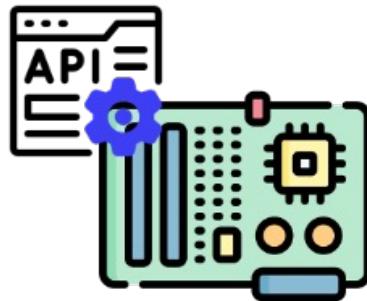
Yapers Web API

```
let changeimg = function(did, yfile_num){  
    get_server_file(did, yfile_num).then((yfile) => {  
        document.getElementById('img').src =  
            `data:image/png;base64, ${yfile}`;  
    });  
};  
setInterval(changeimg, 5000, 4, 1);
```

- `yw_get_data`
- `yw_set_data`
- `yw_get_yfile`
- `yw_set_yfile`
- `yw_get_yvar`
- `yw_set_yvar`
- ...



使用Yapers API
快速開發工作



運行不同節能層級的工作

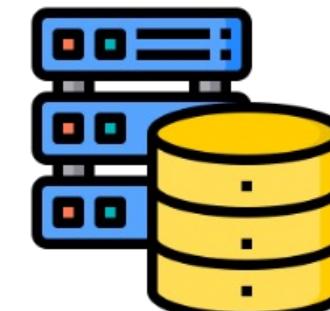
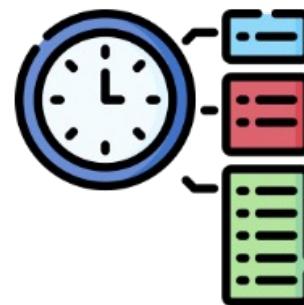
Yapers



使用Web API開發網頁



即時工作排程

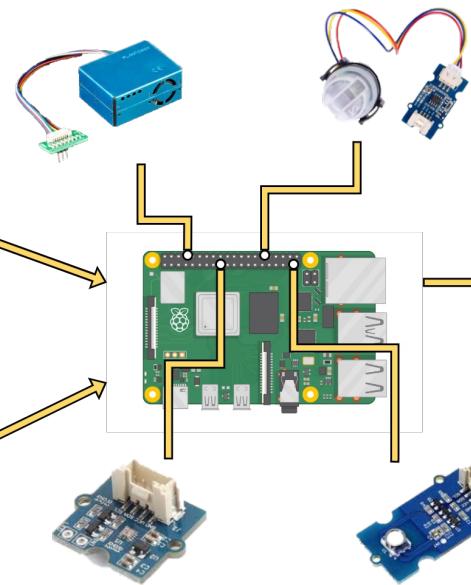


伺服器上的共享資源

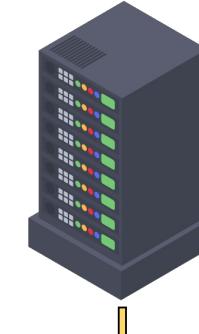


Example#2 : ecoScout

一個用於戶外的生態數據收集器



Yapers Server



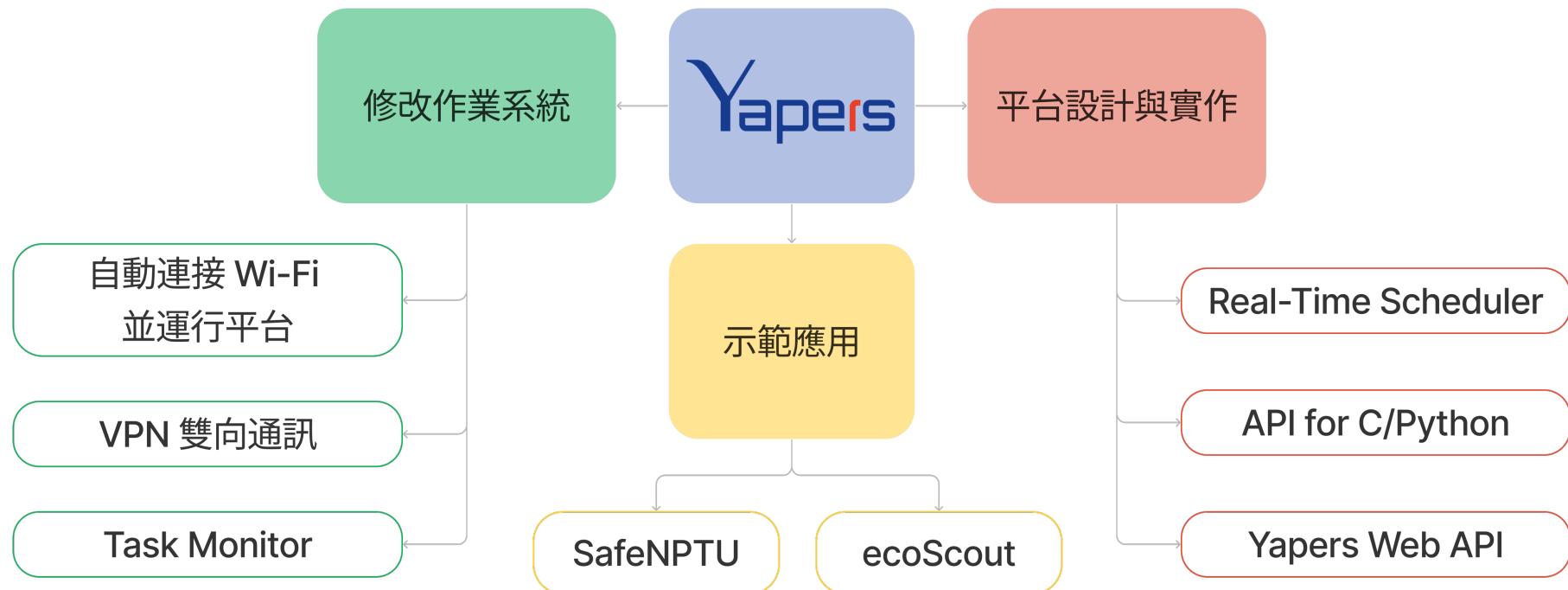
ecoScout Web

名稱	類型
mq9.py	periodic
mq9avg.py	periodic
mq135.py	periodic
mq135avg.py	periodic
SayError.py	sporadic

工作清單



結論





未來展望

- 開發視覺化的 IDE 工具，幫助開發人員更容易地打造各式應用軟體
- 拓展硬體支援性，讓使用 Yapers 開發的軟體可以在更多的硬體上執行
- 強化作業系統的即時性，包含結合 RTAI 以及開發資源存取協定
- 將工作的設計提昇為軟體元件，讓開發人員能夠重複使用既有的工作
- 開放 Yapers 成為 open-source 專案，讓更多人能參與未來的功能開發

簡報到此結束
謝謝聆聽

