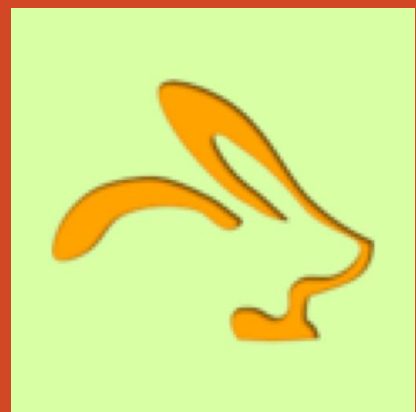


護。邦尼—

遠端動作求援系統

科技與醫護的結合—智慧醫療



108新課綱理想：跨領域素養導向教學

待解決的問題：

- (1)分科教學，學生面對真實問題，無法統整各科知識進行問題解決。
- (2)教材過於強調知識中心，無法與學生的日常生活進行連結。

我們的解法：

(1)教材設計方面

從學生的日常生活進行發想，以學生經驗中存在的事物，將真實世界問題進行簡化，設計一個擬真的專題，融入不同學科知識。

(2)教法設計方面

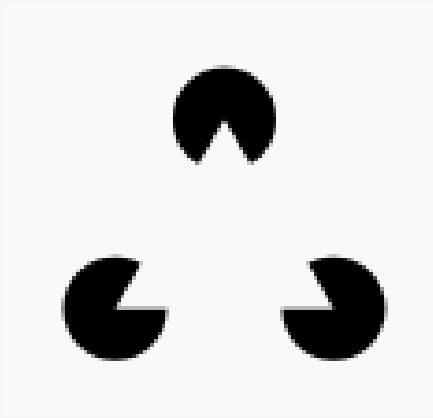
因應未來學生面對的環境強調團隊合作，教學活動設計採用拼圖式合作學習法。將各科知識以科普的方式進行教學，除了培養團隊共同的基礎概念外，也希望進一步讓學生有試探性向的機會，以實現適性化教學。

選定領域與科別

- (1)科技領域：生活科技、資訊科技
- (2)自然領域：物理
- (3)數學領域：數學

教材設計理論依據：格式塔心理學

「整體不同於其部份的總和」



具體性：

雖然左圖沒有畫出三角形，但由於圖片的安排方式，大腦會自動知覺到三角形存在。

教材設計聯想：

- (1)如果教材安排妥當，是否可以讓學生自動知覺到某些知識
- (2)教與不教之間的拿捏，還有如何克服學生個別化的差異

圖片來源：維基百科

應用情境：以學生經驗中的醫院為目標，設計一個擬真的問題，先著重於建立學生對於這個問題的Big Idea

問題描述：如何幫助不方便按鈕呼叫護士的各種不同病患，設計客製化的方式呼叫護士到病床服務？

任務名稱：護。邦尼—遠端動作求援系統

教材重點：AIoT Sensor 智聯感測

感測部份：

Rabboni Sensor with AI programming

應用情境：

客製化病患手勢辨識

科普教學：

將Rabboni手勢辨識簡化為正放與反放

智聯部份：

lotTalk with virtual and physical bulb control

應用情境：

病床與護理站間運用物聯網的真實與虛擬燈泡溝通

科普教學：

運用開發版連結燈泡修改現成的程式連上lotTalk物聯網

分科能力培養

資訊科技相關內涵：Python程式設計

問題 → 解析 → 轉化 → 程式語言 → 結果

生活科技相關內涵：木工、電路學

構想 → 分解 → 施作 → 組合 → 成品

物理相關內涵：電子學

空間 → 抽象 → 公式 → 推導 → 理解

智聯：運用IoTtalk物聯網所見即所得的圖控介面串接連網物

Navigation: s0 Model Flush Delete Simulation ON Import Export

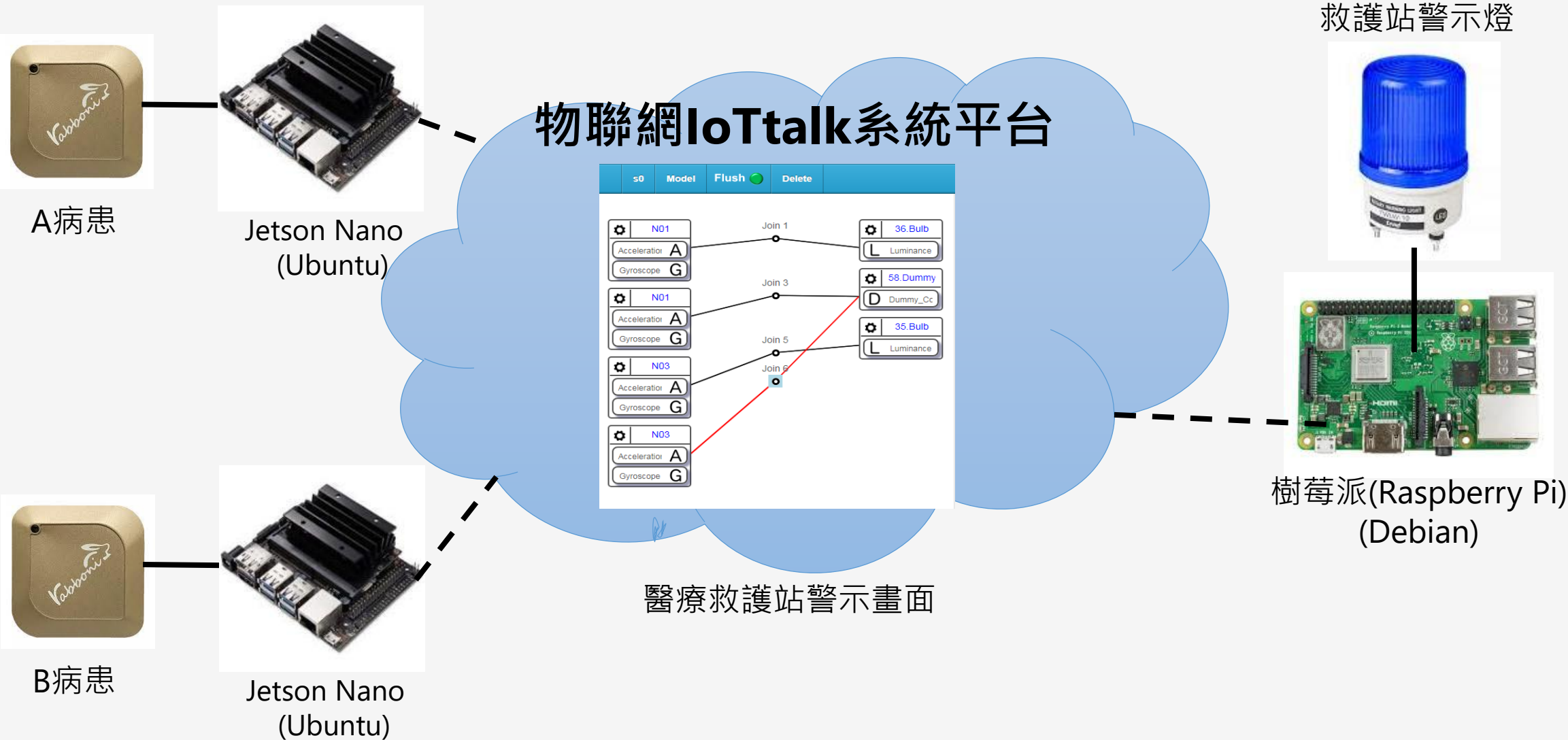
The diagram illustrates a network of IoT devices connected through a central interface. On the left, there are three device blocks labeled N01, N01, and N03. Each block contains 'Acceleratiou A' and 'Gyroscope G' components. On the right, there are three device blocks: 36.Bulb (Luminance), 58.Dummy (Dummy_Cc), and 35.Bulb (Luminance). Connections are labeled as Join 1, Join 3, Join 5, and Join 6. A red line highlights the connection between the bottom N03 device and Join 6.

Connection Name: Join 6 [Delete] [Save]

N03 (IDF) [Delete]		
Acceleration	Type	Function
x1	sample	disabled
x2	sample	
x3	sample	

58.Dummy_Device (ODF) [Delete]	
Dummy_Control	Function
y1	x38

系統架構





感測：使用Python程式將Rabboni六軸感測值讀出並送到IoTtalk



```
DAL.py - C:/Python36/DAL.py (3.6.6)
File Edit Format Run Options Window Help

import time, requests
import RPi.GPIO as GPIO
import DAN
import sys
from rabboni import Rabboni

Rabboni_BLE_address = 'D2:5A:75:76:0D:78' # 要改成你的 Rabboni 的 MAC
ServerURL = 'http://****.iottalk.tw:9999' #with SSL connection # 注意你的 project 所用 Server

Reg_addr = Rabboni_BLE_address #跟 IoTtalk 註冊要用到這

DAN.profile['dm_name']='Smartphone' # 騙 IoTtalk 說我是智慧型手機 :-)
DAN.profile['df_list']=['Acceleration', 'Gyroscope',] #可是我只有六軸感測器
DAN.profile['d_name']='Rabboni02'
DAN.device_registration_with_retry(ServerURL, Reg_addr)

output_pin = 18

# 設置為BCM模式
GPIO.setmode(GPIO.BCM)
# 將腳位設定為輸出，並預設為輸出高電位
GPIO.setup(output_pin, GPIO.OUT, initial=GPIO.HIGH)

Ln: 9 Col: 0
```

```
DAL.py - C:/Python36/DAL.py (3.6.6)
File Edit Format Run Options Window Help

def usb_custom_callback(status):
    global curr_value
    try:
        if status['Acc'][2] > 0 and curr_value==1:
            print("Outputting {} to pin {}".format(curr_value, output_pin))
            GPIO.output(output_pin, curr_value)
            curr_value ^= GPIO.HIGH
            for i in range(5):
                #Push data to an input device feature "Acceleration"
                DAN.push('Acceleration', status['Acc'][0], status['Acc'][1], 9.8)
                DAN.push('Gyroscope', status['Gyr'][0], status['Gyr'][1], status['Gyr'][2])

        if status['Acc'][2] <= 0 and curr_value==0:
            print("Outputting {} to pin {}".format(curr_value, output_pin))
            GPIO.output(output_pin, curr_value)
            curr_value ^= GPIO.HIGH
            for i in range(5):
                #Push data to an input device feature "Acceleration"
                DAN.push('Acceleration', status['Acc'][0], status['Acc'][1], -9.8)
                DAN.push('Gyroscope', status['Gyr'][0], status['Gyr'][1], status['Gyr'][2])

    except Exception as e:
        print(e)
        if str(e).find('mac_addr not found:') != -1:
            print('Reg_addr is not found. Try to re-register...')
            DAN.device_registration_with_retry(ServerURL, Reg_addr)
        else:
            print('Connection failed due to unknow reasons.')
            time.sleep(1)
    except KeyboardInterrupt: # CTRL_C
        try:
            DAN.deregister()
        except Exception as ee:
            print("!!! Try deregister but :", ee)
        finally:
            GPIO.cleanup()
            rabbo.stop() # Rabboni 停止運作
            time.sleep(0.1)

rabbo.start_fetching_status(custom_callback=usb_custom_callback)
rabbo.polling_status()
rabbo.stop_fetching_status()
except Exception as e:
    print(e); print("可能你沒把 Rabboni 用 USB 連接好 ! ")
    sys.exit( )

Ln: 9 Col: 0
```

教法設計策略依據：拼圖式合作學習

專家小組(基礎+進階)



任務小組(科普+統整)

