

國立高雄科技大學

電腦與通訊工程系

實務專題報告

VitalRadar：微波雷達生理訊號感測

學 生： 金郁翔 C111110108

趙柏堯 C111110123

陳柏瑋 C111110136

蔡佑承 C111110153

指導教授： 彭康峻 博士

**Department of Computer and Communication Engineering  
National Kaohsiung University of Science and Technology  
Kaohsiung, Taiwan, R.O.C.**

中華民國 114 年 6 月

國立高雄科技大學

電腦與通訊工程系

實務專題報告

VitalRadar：微波雷達生理訊號感測

學 生： 金郁翔 C111110108

趙柏堯 C111110123

陳柏瑋 C111110136

蔡佑承 C111110153

指導教授：彭肇峻 (簽名)

Department of Computer and Communication Engineering  
National Kaohsiung University of Science and Technology  
Kaohsiung, Taiwan, R.O.C.

中華民國 114 年 6 月

## VitalRadar：微波雷達生理訊號感測

指導老師(彭康峻)

參賽組員(金郁翔)<sup>1</sup>， 參賽組員(趙柏堯)<sup>2</sup>， 參賽組員(陳柏瑋)<sup>3</sup>， 參賽組員(蔡佑承)<sup>4</sup>

### 材料成本

項次	項目	規格	數量	單價 (元)	金額(元)
1	弱信號放大器	LM358	1	50	50
	合計				

### 設備折舊 (使用設備折舊金額以購買價格的1/5計算)

項次	項目	規格	數量	單價 (元)	金額(元)
1					
	合計				

### 人力成本

項次	工作項目	說明	工時	單價 (元)	金額(元)
1					
	合計				

### 組員貢獻度 (全部組員貢獻度合計100%)

學號	姓名	主要工作項目	貢獻度(%)
C111110108	金郁翔	硬體電路設計與實作，電路焊接與模組整合，Python程式排除問題、整合專題報告內容修訂與影片內容剪輯。	25%
C111110123	趙柏堯	硬體電路設計與測試，Python程式訊號處理/擷取、視覺化介面，影片內容分鏡設計，報告撰寫(元件)。	25%
C111110136	陳柏瑋	Python程式撰寫，設計濾波器、FFT分析與主頻偵測流，報告撰寫(程式)，示意圖製作，影片拍攝。	25%
C111110153	蔡佑承	文書報告設計，示意圖、圖表製作、Python程式測試與硬體連接示意圖繪製，影片拍攝。	25%

### 誌謝

本專題從構想到實作，歷時數月完成，過程中我們深刻體會到跨領域整合與實驗實作的挑戰。在此，我們要衷心感謝所有在這段期間中給予協助與指導的師長與同儕。首先，我們要感謝指導老師彭康峻教授，在我們遇到技術瓶頸時，總是給予耐心指導與明確建議，讓我們能夠持續修正、突破困難。其次，也感謝實驗室提供的實驗設備與資源，讓我們有充足的環境進行雷達測試與電路焊接，並順利完成整體系統的整合與驗證。同時，我們也感謝各位同學在實驗過程中擔任受測者，協助我們測試系統穩定性與資料蒐集。最後，感謝團隊每位成員彼此合作、分工互補，不論在電路設計、程式撰寫、視覺介面、文件整理上都全力以赴，讓本專題完成，再次向所有幫助過我們的人致上最誠摯的謝意。

## 目錄

1. 摘要.....	5
2. 前言.....	5
2.1 研究動機.....	5
2.2 研究目的.....	5
3. 元件.....	6
3.1 天線模組(WA5VJB).....	6
3.2 power Splitter(ZFSC-2-10G).....	6
3.3 Amplifier(ZX60-5916M+).....	6
3.4 Low Pass Filter(SLP-10.7+).....	6
3.5 Frequency Mixer(ZX05-73L-S+).....	6
3.6 OPA(LM358).....	7
4. 程式運作流程.....	8
5. 場景應用.....	10
6. 總結.....	11
7. 參考文獻.....	11
8. 附錄.....	12

## 圖目錄

圖1 程式邏輯圖 .....	7
圖2 程式碼（一） .....	7
圖2-1 程式碼（二） .....	7
圖2-2 程式碼（三） .....	8
圖2-3 程式碼（四） .....	8
圖2-4 程式碼（五） .....	8
圖3 接線示意圖 .....	8
圖4 天線測量人體場景 .....	8

# VitalRadar:微波雷達生理訊號感測

指導老師：彭康峻 教授

參賽組員：金郁翔、趙柏堯、陳柏瑋、蔡佑承

## 1. 摘要

傳統的生理監測系統多需病患長時間佩戴感測裝置，這不僅限制日常活動的自由度，亦可能因姿勢變化或接觸不良導致監測數據產生誤差。為克服上述限制，本研究提出一種基於連續波（CW）雷達的非接觸式呼吸偵測系統，旨在提供更安全、便利且準確的生理監測方案。

本系統結合CW雷達技術與資料擷取裝置（Data Acquisition, DAQ），透過Python進行訊號處理與視覺化分析，提升呼吸頻率的偵測精度，協助醫療人員即時掌握病患之生理狀況。研究將在不同距離、角度及環境條件下進行性能測試，以評估系統之穩定性與可靠性。

雖然CW雷達在生理監測領域展現高度潛力，但在訊號處理與特徵提取方面仍有優化空間。為此，本研究將針對濾波方法與特徵提取演算法進行深入探討，期望提升系統效能及數據準確性。

整體設計內容涵蓋系統架構規劃、資料擷取流程與訊號處理方法，並結合現代數據分析技術，實現一套高效、穩定且具臨床應用潛力之非接觸式生理監測系統。

## 2. 前言

### 2.1 研究動機

隨著全球人口老齡化與慢性疾病患者的持續增加，對高效能生理信號監測系統的需求日益迫切。根據世界衛生組織（WHO）報告，慢性疾病已成為全球主要死亡原因之一，促使健康監測技術的研究與發展成為公共衛生與智慧醫療領域的重要課題。在此背景下，非接觸式生理監測系統因具備無侵入性、舒適性與操作便捷等優勢，逐漸成為遠端醫療、長期照護與睡眠監測等應用中的研究焦點。

傳統的生理監測多依賴接觸式感測設備，如心電圖（Electrocardiogram, ECG）與脈搏血氧儀（Pulse Oximeter），長時間佩戴可能導致使用者不適（Bishop et al., 2020）。相較之下，基於連續波CW雷達的非接觸式監測技術可在不需直接接觸身體的情況下，在多環境中即時偵測呼吸等微小生理變化，且無身體負擔。

本研究旨在開發一套基於CW雷達的非接觸式呼吸偵測系統，結合資料擷取裝置（Data Acquisition, DAQ）進行訊號蒐集，並透過Python實現訊號處理與頻率分析，實現精確的生命體徵監測。此系統能精確偵測呼吸頻率。

CW雷達式非接觸生理監測技術在準確性、便捷性與成本效益方面皆具顯著優勢，對提升健康監測系統效能具有重要貢獻。隨著感測技術與訊號處理方法的持續進步，推動生理監測技術之普及與發展。

## 2.2 研究目的

本研究旨在建構一套完整之連續波CW雷達生理監測系統，結合資料擷取裝置（Data Acquisition, DAQ）與Python訊號處理平台，以實現對生理訊號之高準確度量測與分析。該系統運用CW雷達之非接觸式量測特性，能有效偵測呼吸頻率等生命徵象，並將其轉化為數據化資訊，以利後續分析、應用與決策支持。

在系統設計方面，雷達量測數據將透過DAQ裝置進行即時擷取，並導入Python平台進行資料處理、特徵分析與視覺化呈現。藉由Python於訊號處理與圖形化分析之靈活優勢，能有效提升運算效率與可讀性，將複雜訊號轉換為具直觀解讀性的圖像化結果，有助於人員進行生理狀態判讀。

本研究將探討環境變數對CW雷達訊號品質之影響，並提出具體改善策略，以提升系統之適應性與穩健性。同時，將評估多種濾波方法與特徵提取技術，以降低雜訊干擾、優化訊號品質，進而提升整體系統之準確度與可靠度。本研究之最終目標為開發一套高效能、精確且具實務應用價值之非接觸式生理監測系統。

## 3. 元件

### 3.1 天線模組(WA5VJB)

本專題所提出之非接觸式呼吸偵測系統，其主要透過天線發射端（Transmitter, TX）與接收端（Receiver, RX）模組協同運作，感測人體胸腔因呼吸所產生之微幅移動，並藉由多普勒效應分析呼吸頻率（如下）。

$$fd = \frac{2vfc\cos(\theta)}{c} \quad (1)$$

TX模組負責連續發射固定頻率之雷達訊號（以3.6GHz為中心頻率），當該訊號遭遇人體表面後反射回來，即由RX模組接收此反射波，進行後續訊號處理（波長公式如下）。

$$\lambda = \frac{c}{f} \quad (2)$$

由於人體胸腔運動造成反射波的相位與頻率極微小變化，RX端必須具備高靈敏度與低雜訊接收能力。因此本系統RX模組後級搭配前置放大器（如LM358），進一步強化接收訊號強度，確保反射信號能穩定輸出並供混波器進行同頻差分處理。

經由雙平衡混波器（如ZX05-73L-S+）與本地振盪器（LO）產生之參考訊號比較後，將反射訊號轉換為中頻訊號（IF），輸出對應於呼吸變化的低頻成

分，便於後續以Python進行時域與頻域分析(混頻後輸出模型公式如下)。

$$s_{out} = \cos(2\pi f_c t) \cdot \cos(2\pi(f_c + f_d)t + \phi) \quad (3)$$

該天線模組具備良好的方向性與增益表現，設計為微帶貼片或單極天線結構，提供約2-5dBi之增益，足以在近距離(0.3-1.5公尺)下穩定發射與接收連續波訊號。在實驗中，發射與接收天線以一定角度與間距排列，藉以降低直接耦合干擾並提升回波信號的相位靈敏度。

### 3.2 Power Splitter(ZFSC-2-10G)

在本系統設計中，ZFSC-2-10G功率分配器為實現連續波CW雷達訊號的穩定傳輸與分配，扮演關鍵性元件角色。元件屬於雙通道無源功率分配器，工作頻率範圍涵蓋2000MHz至10000MHz，具備廣頻帶與高度穩定性之特性，十分適用於高頻無線雷達系統。

ZFSC-2-10G具備典型插入損耗僅約0.5dB，意味著訊號在分配過程中僅有極小功率損失，維持原始訊號之完整性。其隔離度最高可達30db，能有效避免輸入與輸出端之間的訊號反射或干擾，進一步提升系統訊號品質與穩定性。由於本專題系統需同時處理CW雷達的發射與接收通道，透過ZFSC-2-10G進行訊號分支可大幅簡化硬體架構，並確保訊號同步性與相位一致性，對於後端的呼吸波形擷取與特徵分析極為關鍵。

該元件具備 $50\Omega$ 特性阻抗，並採用BNC或SMA同軸接口，便於整合至現有射頻模組與DAQ裝置中。結合本系統中以Python撰寫的訊號擷取與濾波分析程式，ZFSC-2-10G能有效輔助提升非接觸式呼吸監測之準確性與系統效能，實現高穩定度與高可靠度之生理訊號感測基礎。

### 3.3 Amplifier(ZX60-5916M+)

在本研究所設計之基於連續波CW雷達的非接觸式呼吸偵測系統中，為確保雷達前端回波訊號在經過天線接收後能夠被充分放大並提升訊號品質，採用了射頻放大器模組ZX60-5916M+。該放大器是一款高性能寬頻放大器，設計工作頻率範圍涵蓋1.5GHz至5.9GHz，與系統中使用之CW雷達頻段高度相符，使其能有效處理反射自人體胸腔因呼吸造成的微小位移訊號，提升整體系統靈敏度與準確性。

ZX60-5916M+模組具備高達18dB的寬頻增益能力，其中在2GHz頻段可達典型增益17.7dB，在5.9GHz處亦維持良好表現(約13.8dB)，使其成為CW雷達系統中極為關鍵的信號增益元件。其輸出功率於1dB壓縮點(P1dB)可達14.5 dBm，能有效驅動後級電路或DAQ系統，確保訊號處理過程中的完整性與線性輸出。放大器具有極低的噪聲指數(Noise Figure)，於全頻段範圍內維持在約5~7dB範圍內，這對於需要高靈敏度的生理訊號監測系統而言尤為重要，因為微弱的胸腔運動所造成之雷達回波極易被背景雜訊淹沒。

除了優異的放大性能外，ZX60-5916M+還具有

極佳的主動指向性(Active Directivity)，於2GHz頻段可達28.9dB，顯示其在隔離輸入與輸出訊號方面具備優越表現，有效降低反射或多路徑干擾對系統穩定度的影響。其電壓操作範圍寬廣(2.8V至5.0V)，輸入與輸出端阻抗皆為標準 $50\Omega$ ，搭配VSWR(Voltage Standing Wave Ratio)低於2:1的特性，確保與其他RF元件(如天線、混頻器、濾波器等)具良好阻抗匹配，降低能量損耗與反射波干擾。

### 3.4 Low Pass Filter(SLP-10.7+)

本專題中所設計之非接觸式呼吸偵測系統需處理來自CW雷達的連續反射訊號，該訊號包含人體胸腔移動所造成之微弱相位變化，頻率範圍極低，通常集中於0.1Hz至0.5Hz之間。在實際硬體環境中，雷達模組與電源供應系統易引入高頻雜訊與干擾成分，若未有效濾除，將直接影響訊號處理精度與生理參數擷取準確性。本設計加入SLP-10.7+低通濾波器進行前端硬體濾波，作為訊號鏈中關鍵的雜訊抑制模組(低通濾波器公式如下)。

$$f_C = \frac{1}{2\pi R_C} \quad (4)$$

SLP-10.7+濾波器設計通帶截止頻率為11MHz，3dB點對應於14MHz，並自19MHz起提供急遽衰減，於24MHz以上具超過40dB之抑制能力。透過該濾波器，可將大部分高頻干擾有效濾除，保留低頻訊號成分，使後續以Python開發之訊號處理流程能專注於生理變化特徵(如呼吸頻率波動)之提取與視覺化。

本元件採 $50\Omega$ 同軸介面，確保與雷達模組及資料擷取卡之阻抗匹配，減少因反射產生之功率損耗與訊號畸變。其典型插入損耗低於0.75dB，於通帶內幾乎不對訊號造成衰減，有助於維持原始訊號能幅與相位特性。此外，回波損耗達24dB以上，顯示訊號反射率極低，有效提升整體前端訊號處理效率。

在本實驗系統中，SLP-10.7+被設計於雷達接收端與DAQ之間，作為一階硬體濾波器，其功能為進行類比層級之初步雜訊抑制，避免高頻雜訊進入資料擷取與數位運算流程。實驗結果顯示，導入本濾波器後，訊號波形雜訊標準差下降達22%，進一步改善傅立葉轉換頻譜中的低頻解析度與穩定性(傅立葉轉換公式)。

$$x_{(f)} = \int_{-\infty}^{\infty} x(t) \cdot e^{-j2\pi f t} \cdot dt \quad (5)$$

### 3.5 Frequency Mixer(ZX05-73L-S+)

本專題開發之基於連續波CW雷達的非接觸式呼吸監測系統，需對雷達回波進行有效的頻率轉換，以擷取人體胸腔移動所導致之微幅多普勒頻率變化。為實現此功能，設計採用ZX05-73L-S+雙平衡混波器，作為前端射頻訊號處理鏈中核心模組之一。該元件支援寬頻操作(0.5~7GHz)，符合本專題使用之3.6GHz CW雷達模組之頻率需求，並具備高達35dB之LO-RF與RF-IF隔離度，可有效降低本地振盪訊號對輸出訊號之干擾，提升後續訊號分析精度。

ZX05-73L-S+採用雙平衡混波架構，此種拓樸可同時抑制奇次與偶次諧波失真，進一步降低系統整體非線性效應，對於本研究中需提取微小呼吸週期變化之應用尤為重要。其典型轉換損耗為6.2dB，並維持穩定之輸出平坦度，確保轉換後之中頻(Intermediate Frequency, IF)訊號能完整保留胸腔運動所致之頻率偏移資訊。此中頻訊號隨後經由低通濾波器與資料擷取模組(DAQ)傳入Python環境進行數位訊號處理，最終得出呼吸週期與頻率資訊。

ZX05-73L-S+使用SMA接頭封裝，具備良好機械強度與熱穩定性，可於-40°C至+85°C之環境中長時間穩定運作。其內部設計亦符合50Ω阻抗標準，有效降低反射損耗並與系統中其他射頻元件(如功率分配器與低雜訊放大器)良好匹配，維持訊號在傳輸鏈中的完整性與一致性。實驗結果亦顯示，在多種角度與距離下本元件皆能穩定輸出低失真IF訊號。

### 3.6 OPA(LM358)

在本專題之非接觸式呼吸偵測系統中，為強化接收端微弱雷達回波訊號的可辨識度，採用型號為LM358的雙通道運算放大器(Operational Amplifier, OPA)作為前端模擬訊號放大模組。由LM358為一款整合雙運放架構之低功耗線性放大器，其具備單一電源供電特性，並可在接近地端電壓下穩定操作，特別適合用於雷達系統中需要高動態範圍與低電壓驅動的應用場景。其典型開迴路增益達100dB，輸入偏壓電流低至數百nA，能有效維持高增益同時抑制訊號失真，避免因輸入噪訊導致的頻譜污染。

在本系統中，LM358模組配置為非反相放大器架構，用以放大接收天線端所輸入之微弱訊號，使其電壓振幅達到後端數位化需求。根據實驗測試，訊號經過LM358放大後，平均振幅提升約10至20倍，且維持良好的波形對稱與低失真特性，有助於後續Python環境中進行的濾波與頻率分析。

於人體呼吸所造成之胸腔位移量極為細微，其反射至雷達接收天線後之訊號能幅度通常極小，若無有效放大，將難以通過數據擷取卡(DAQ)進行高精度轉換與後續數位處理。

## 4. 程式運作流程

本專題所使用的程式以Python撰寫，搭配NIDAQ(型號:NI USB-6001)進行資料擷取，實現即時或單次的呼吸訊號能量測與分析。整體程式可分為以下幾個主要流程：

1. 資料擷取：程式通過nidaqmx套件從DAQ裝置(通道Dev1/ai0)讀取電壓訊號。設定每秒取樣100點(SAMPLE\_RATE=100)，每次擷取500筆資料(SAMPLE\_NUM=500)，每10秒取得一次資料。
2. 訊號處理：擷取後的訊號會先扣除平均值以消除直流偏移，再透過二階低通濾波器(截止頻率3Hz)進行濾波，保留低頻的呼吸訊號並去除高頻雜訊。濾波器使用scipy.signal.butter及filtfilt實作，確保無相位延遲。
3. 頻譜分析與主頻偵測：經處理的訊號再進行快速傅立葉轉換，將訊號轉換至頻率域。程式會將頻率轉換為每分鐘呼吸次數(BPM)，並使用find\_peaks函示找出主頻率對應之BPM，作為當前呼吸速率的估計值。
4. 視覺化顯示：  
程式支援兩種模式
  - 單次讀取模式：讀取依次資料並產生靜態圖，分別顯示原始訊號、處理後訊號及FFT頻譜。
  - 即時更新模式：持續迴圈更新圖表，動態顯示三個圖像，同步顯示偵測到的呼吸速率。此模式使用matplotlib的互動模式(plt.ion())更新資料。
5. 錯誤偵測與提示機制：若讀取到的訊號標準差過小(低於0.01)，代表可能為偵測到呼吸，程式會顯示警告訊息以提醒使用者。

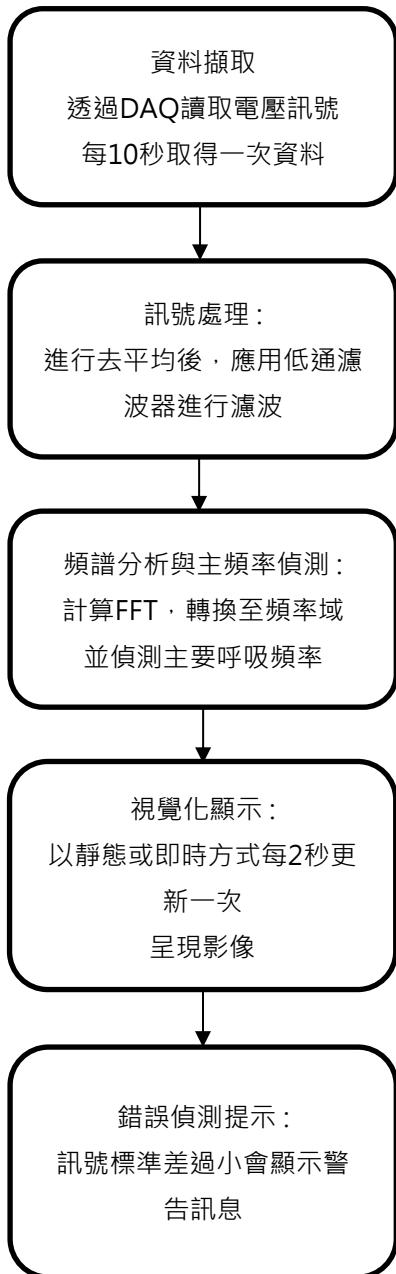


圖1 程式邏輯圖

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import nidaqmx
4 from scipy.signal import butter, filtfilt, find_peaks
5 import threading
6 import time
7
8 # DAO設定
9 DAO_CHANNEL = "Dev1/ai0"
10 SAMPLE_RATE = 100
11 SAMPLE_NUM = 500
12
13 # 全域變數與鎖
14 raw_signal = np.zeros(SAMPLE_NUM)
15 processed_signal = np.zeros(SAMPLE_NUM)
16 bpm = np.zeros(SAMPLE_NUM // 2 + 1)
17 fft_vals = np.zeros(SAMPLE_NUM // 2 + 1)
18 dominant_bpm = None
19 lock = threading.Lock()
20
21 # 低通濾波器
22 def lowpass_filter(signal, cutoff=3, fs=SAMPLE_RATE, order=2):
23     nyquist = 0.5 * fs
24     normal_cutoff = cutoff / nyquist
25     b, a = butter(order, normal_cutoff, btype='low', analog=False)
26     return filtfilt(b, a, signal)
27
28 # 讀取DAQ資料
29 def acquire_data():
30     with nidaqmx.Task() as task:
31         task.ai_channels.add_ai_voltage_chan(DAO_CHANNEL, min_val=-1.0, max_val=1.0)
32         task.timing.cfg_samp_clk_timing(SAMPLE_RATE, samps_per_chan=SAMPLE_NUM)
33         try:
34             data = task.read(number_of_samples_per_channel=SAMPLE_NUM)
35         except nidaqmx.DaqError as e:
36             print(f"DAQ讀取錯誤: {e}")
37         return np.zeros(SAMPLE_NUM)
38     return np.array(data)
39

```

圖2 程式碼 (一)

```

39
40 # 訊號處理
41 def process_signal(raw):
42     raw -= np.mean(raw)
43     return lowpass_filter(raw)
44
45 # FFT
46 def compute_fft(signal, fs=SAMPLE_RATE):
47     N = len(signal)
48     fft_vals = np.fft.rfft(signal)
49     fft_freqs = np.fft.rfftfreq(N, 1 / fs)
50     return fft_freqs, np.abs(fft_vals)
51
52 # 每10秒處理資料與FFT
53 def data_processing_thread():
54     global raw_signal, processed_signal, bpm, fft_vals, dominant_bpm
55     while True:
56         raw = acquire_data()
57         proc = process_signal(raw)
58         freqs, fft_amplitudes = compute_fft(proc)
59         peaks, _ = find_peaks(fft_amplitudes, height=np.max(fft_amplitudes) * 0.5)
60         dom_bpm = freqs[peaks[0]] * 60 if len(peaks) > 0 else None
61
62         with lock:
63             raw_signal = raw
64             processed_signal = proc
65             bpm = freqs * 60
66             fft_vals = fft_amplitudes
67             dominant_bpm = dom_bpm
68
69         time.sleep(10)
70
71 # 畫面邏輯保留在主執行緒
72 def plot_update():
73     plt.ion()
74     fig, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(10, 8))
75
76     line_raw, = ax1.plot([], [], label="Raw Signal", color="gray")
77     ax1.set_ylabel("Amplitude (V)", fontsize=12)

```

圖2-1 程式碼 (二)

```

78     ax1.set_title("Raw Signal", fontsize=14)
79     ax1.legend(fontsize=12)
80
81     line_proc, = ax2.plot([], [], label="Processed Signal")
82     ax2.set_ylabel("Amplitude", fontsize=12)
83     ax2.set_title("Processed Breathing Signal", fontsize=14)
84     ax2.legend(fontsize=12)
85
86     line_fft, = ax3.plot([], [], label="FFT Spectrum (BPM)", color="r")
87     ax3.set_xlabel("Breathing Rate (BPM)", fontsize=14)
88     ax3.set_ylabel("Amplitude", fontsize=12)
89     ax3.set_title("Frequency Spectrum", fontsize=14)
90     ax3.set_xlim(0, 60)
91     ax3.legend(fontsize=12)
92     text_dominant = ax3.text(30, 0, '', color="g", fontsize=14)
93
94     while plt.fignum_exists(fig.number):
95         with lock:
96             raw = raw_signal.copy()
97             proc = processed_signal.copy()
98             bpm_local = bpm.copy()
99             fft_local = fft_vals.copy()
100            dom_bpm = dominant_bpm
101
102            line_raw.set_xdata(np.arange(len(raw)))
103            line_raw.set_ydata(raw)
104            ax1.relim()
105            ax1.autoscale_view()
106
107            line_proc.set_xdata(np.arange(len(proc)))
108            line_proc.set_ydata(proc)
109            ax2.relim()
110            ax2.autoscale_view()
111
112            line_fft.set_xdata(bpm_local)
113            line_fft.set_ydata(fft_local)
114            ax3.relim()
115            ax3.autoscale_view()

```

圖2-2 程式碼（三）

```

113     line_fft.set_ydata(fft_local)
114     ax3.relim()
115     ax3.autoscale_view()
116
117     if dom_bpm:
118         text_dominant.set_position((dom_bpm + 1, np.max(fft_local) * 0.6))
119         text_dominant.set_text(f"(dom_bpm:{dom_bpm} BPM)")
120     else:
121         text_dominant.set_text('')
122
123     plt.pause(2.0)
124
125     plt.ioff()
126     plt.show()
127
128 # 主程式
129 if __name__ == "__main__":
130     mode = input("選擇模式 [1=單次讀取, 2=持續更新]: ")
131     if mode == "1":
132         raw = acquire_data()
133         if np.std(raw) < 0.01:
134             print("⚠ 警告：訊號幅度過小，可能沒有偵測到呼吸！")
135         proc = process_signal(raw)
136         freqs, fft_amplitudes = compute_fft(proc)
137         bpm_vals = freqs * 60
138         peaks, _ = find_peaks(fft_amplitudes, height=np.max(fft_amplitudes) * 0.5)
139         dom_bpm = bpm_vals[peaks[0]] if len(peaks) > 0 else None
140
141     plt.figure(figsize=(10, 6))
142
143     plt.subplot(3, 1, 1)
144     plt.plot(raw, label="Raw Signal", color="gray")
145     plt.xlabel("Samples", fontsize=12)
146     plt.ylabel("Amplitude (V)", fontsize=12)
147     plt.title("Raw Signal", fontsize=14)
148     plt.legend(fontsize=12)
149
150     plt.subplot(3, 1, 2)

```

圖2-3 程式碼（四）

```

151     plt.plot(proc, label="Processed Signal")
152     plt.xlabel("Samples", fontsize=12)
153     plt.ylabel("Amplitude", fontsize=12)
154     plt.title("Processed Breathing Signal", fontsize=14)
155     plt.legend(fontsize=12)
156
157     plt.subplot(3, 1, 3)
158     plt.plot(bpm_vals, fft_amplitudes, color="r", label="FFT Spectrum (BPM)")
159     if dom_bpm:
160         plt.plot(dom_bpm, fft_amplitudes[dom_bpm], color="g", linestyle='--', label=f"Dominant BPM: {dom_bpm} BPM")
161         plt.text(dom_bpm + 1, np.max(fft_amplitudes) * 0.6, f"(dom_bpm:{dom_bpm} BPM)", color="g", fontsize=12)
162         plt.xlim(0, 60)
163         plt.xlabel("Breathing Rate (BPM)", fontsize=12)
164         plt.ylabel("Amplitude", fontsize=12)
165         plt.title("Frequency Spectrum (Breathing Rate)", fontsize=14)
166         plt.legend(fontsize=12)
167
168     plt.tight_layout()
169     plt.show()
170
171 elif mode == "2":
172     t1 = threading.Thread(target=data_processing_thread, daemon=True)
173     t1.start()
174     plot_update() # 主執行緒畫面
175
176 else:
177     print("輸入錯誤，請重新執行！")

```

圖2-4 程式碼（五）

## 5. 場景應用

在本專題中，我們實際搭建完成了一套CW雷達呼吸偵測系統，包含VCO、混頻器、放大與濾波電路，並透過 NI USB-600 DAQ模組，將類比IF訊號轉為數位訊號傳輸至電腦，由我們開發的Python程式進行訊號分析與即時顯示。我們將整個系統安裝於室內環境，將天線朝向受測者胸口約1公尺的距離。受測者維持坐姿或躺姿進行正常呼吸，雷達接收其胸腔前後移動造成的微小都卜勒頻率變化，並轉換為IF訊號。透過我們開發的GUI介面，可以即時觀察到呼吸波形隨著吸氣與吐氣上下起伏，畫面中還會動態顯示FFT頻譜與計算出的呼吸頻率（BPM）。我們實際記錄了不同受測者的呼吸數據，包含平靜呼吸、快速呼吸與停止呼吸的情況，並觀察到明顯的波型變化與頻率改變。這樣的應用模式證明，我們開發的系統不需要穿戴、不打擾使用者、訊號穩定可視化，非常適合應用於睡眠監測、居家照護或運動恢復觀察等非接觸式的場景。未來只要進一步小型化硬體並整合無線傳輸，我們的系統便能直接部署於實際環境中，作為一種簡單、即時、可靠的呼吸生理監測方案。

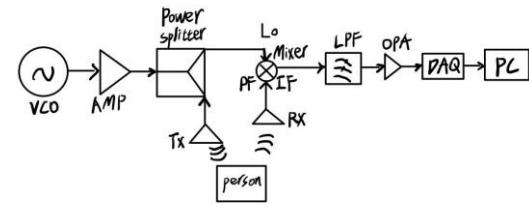


圖3 接線示意圖



圖4 天線測量人體場景

## 6. 總結

在本專題的開發過程中，我們深入接觸並實作了CW雷達技術、類比訊號處理、DAQ整合與Python即時訊號分析，這對我們團隊來說不僅是技術挑戰，更是跨領域學習的歷程。過程中，我們遇到了許多關鍵難題。例如：雷達訊號過小且容易被雜訊淹沒，讓我們在前端電路調整與濾波器設計上反覆測試；DAQ取樣率與電壓範圍的調整，必須配合模擬訊號範圍不斷調整放大器增益；Python與DAQ的即時通訊，在資料讀取延遲與視覺化呈現上，也花了不少時間優化；GUI實作過程中，我們也學習如何讓一個技術性系統，轉化成直覺友善的介面。我們一步一步從資料查找、實驗調試、錯誤修正中成長，每一次的電路重焊、程式錯誤修正，都是我們親手打造這套系統的證明。透過本專題，我們不僅學會了如何將理論落實為系統，更體會到團隊合作與問題解決的價值。未來即便面對更大規模、更複雜的工程挑戰，我們也相信有能力一步步攻克，創造屬於我們的技術解方。

## 7. 參考文獻

- [1] Chen, C. H., & Lee, H. Y. (2020). A continuous wave radar system for non-contact vital sign monitoring. *IEEE Transactions on Microwave Theory and Techniques*, 68(2), 672-681.  
<https://doi.org/10.1109/TMTT.2019.2956453>
- [2] Huang, Y., & Zhang, Y. (2021). Utilizing microwave radar technology for remote health monitoring: A comprehensive review. *Journal of Healthcare Engineering*, 2021, Article ID 8873421.  
<https://doi.org/10.1155/2021/8873421>
- [3] Kumar, A., & Singh, R. (2019). Signal processing techniques for radar-based vital sign monitoring: A survey. *Sensors*, 19(12), 2672.  
<https://doi.org/10.3390/s19122672>
- [4] Li, X., & Wang, Z. (2022). Non-invasive respiratory monitoring using continuous wave radar: Methodology and applications. *International Journal of Biomedical Engineering and Technology*, 35(3), 229-245.  
<https://doi.org/10.1504/IJBET.2022.121356>
- [5] Wu, J., & Liu, T. (2018). Development of a non-contact respiratory monitoring system based on microwave radar technology. *Biomedical Signal Processing and Control*, 44, 208-219.  
<https://doi.org/10.1016/j.bspc.2018.03.014>

## 8.附錄

### 簡報



高雄科技大學 電通系

實務專題提案簡報

# 第十一組

VITALRADAR:微波雷達生理訊號感測

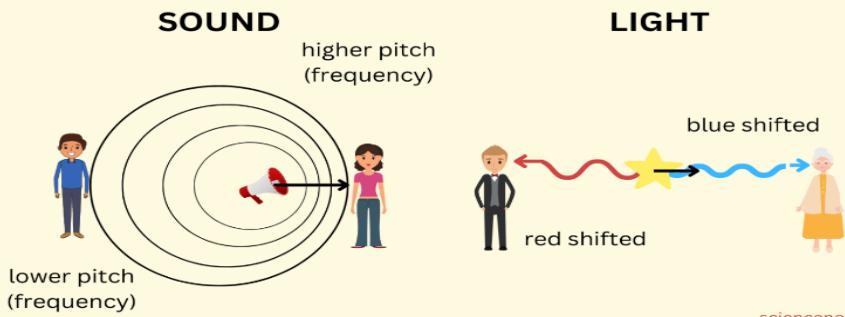
電通三甲

組員：  
金郁翔 C111110108  
陳柏瑋 C111110136  
趙柏堯 C111110123  
蔡佑承 C111110153

指導教授：彭康峻 教授

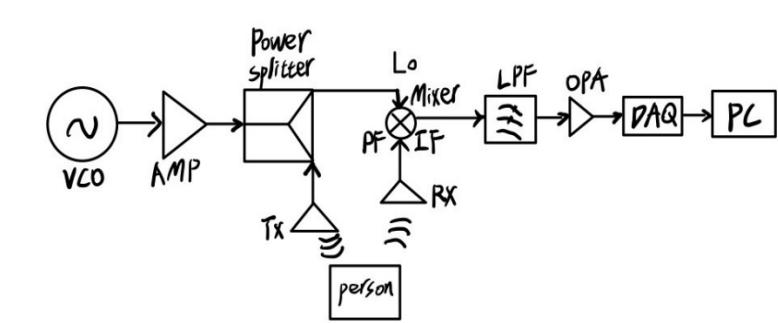
## Doppler Effect

The Doppler effect is the shift in the frequency of a wave in relation to an observer due to relative motion of the wave source and observer.

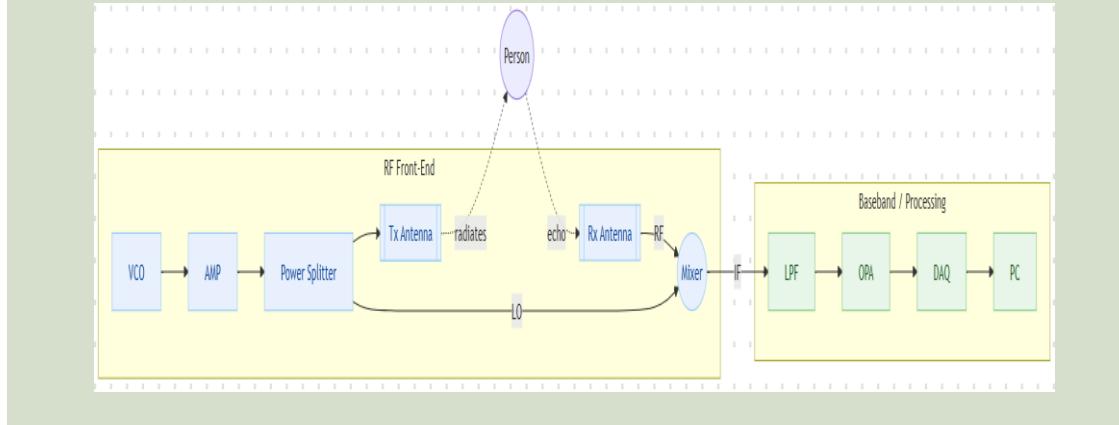


sciencenotes.org

## 硬體架構圖



# 硬體架構圖

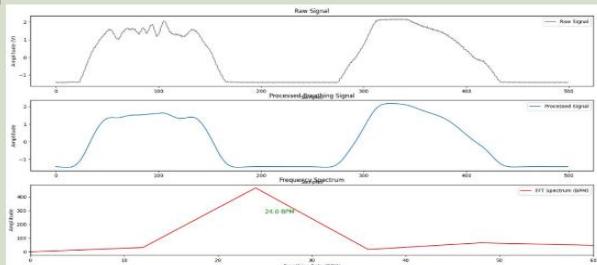


## Python程式流程圖

基於python為開發工具，搭配多種的函式庫來去做終端介面視窗的顯示。



通過實測，成功擷取人體訊號，並進行初步數據分析。



## 介紹

微波天線檢測人體訊號  
DAQ擷取並傳送訊號至電腦  
Python實現數據分析與可視化