

# 跌倒警示裝置

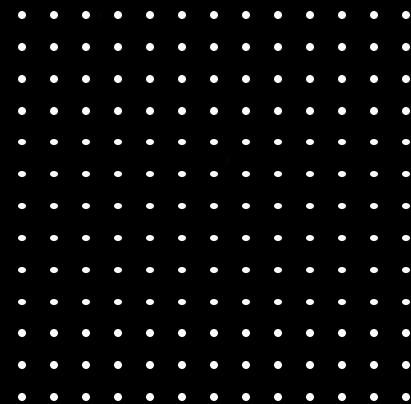
學校：三民國中

指導老師：魏子寰

組員：楊昊展、賴侑廷、吳黃睿

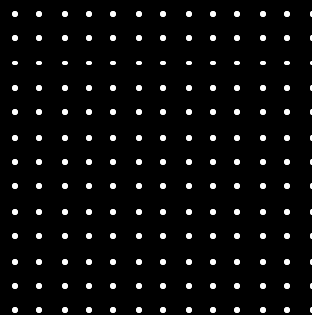
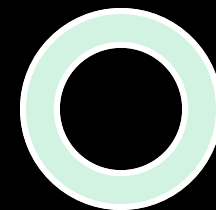
編號：00050

Robboni應用



# 大綱

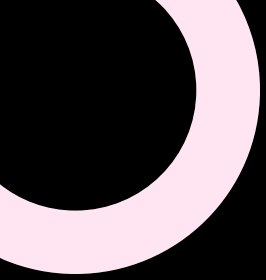
- 設計理念/原因
- 關於跌倒警示裝置
- 實作方式
- 實際展示
- 未來展望



# 設計理念/原因

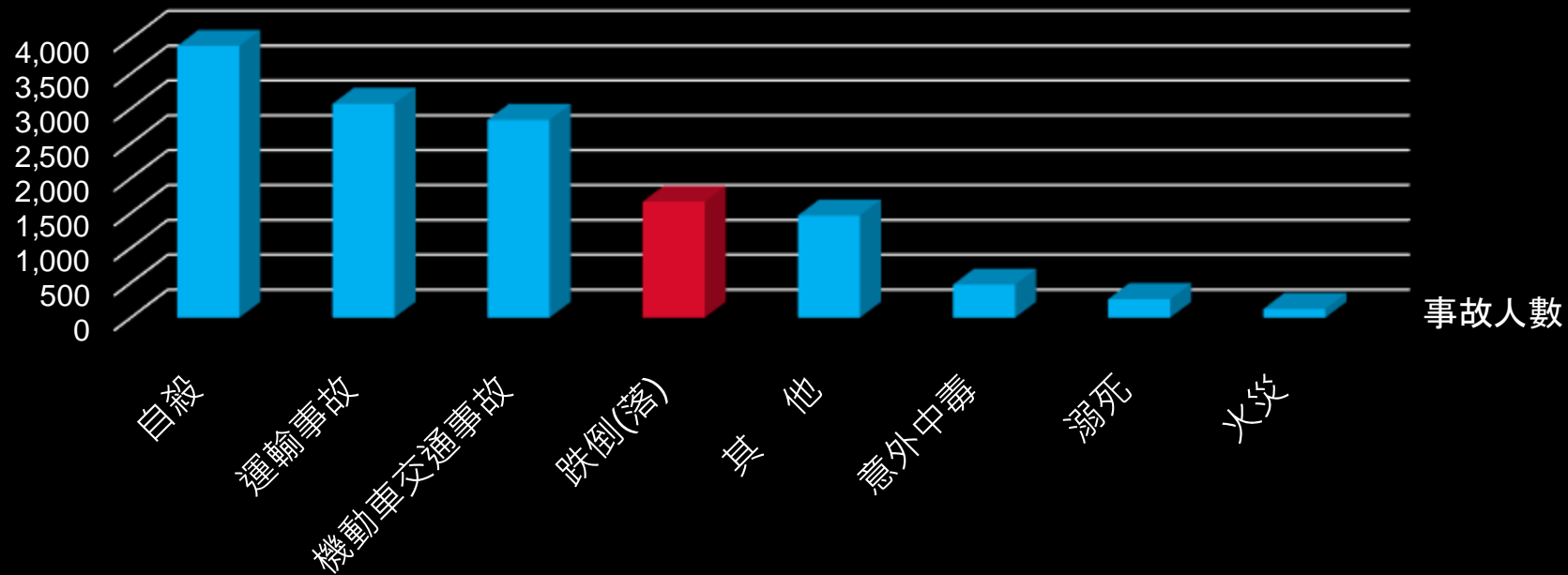
在資訊課老師教我們使用Rabboni進行六軸感測時，我們想起先前我們在新聞上看到許多人因跌倒且未即時受到即時的救治，導致不幸離世，在我們查閱相關資料後發現，事故傷害死亡中排名第4，僅次於車禍事故，於是我們希望可以設計一款可以偵測跌倒並即時通知家人的裝置，以免憾事再次發生。





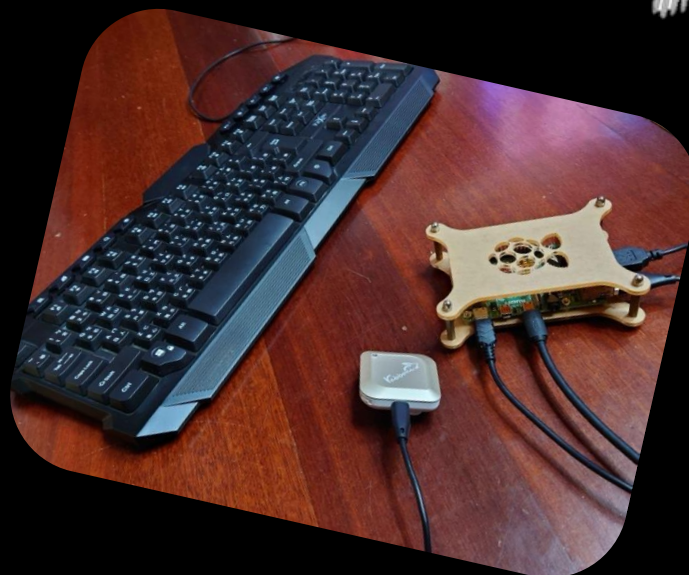
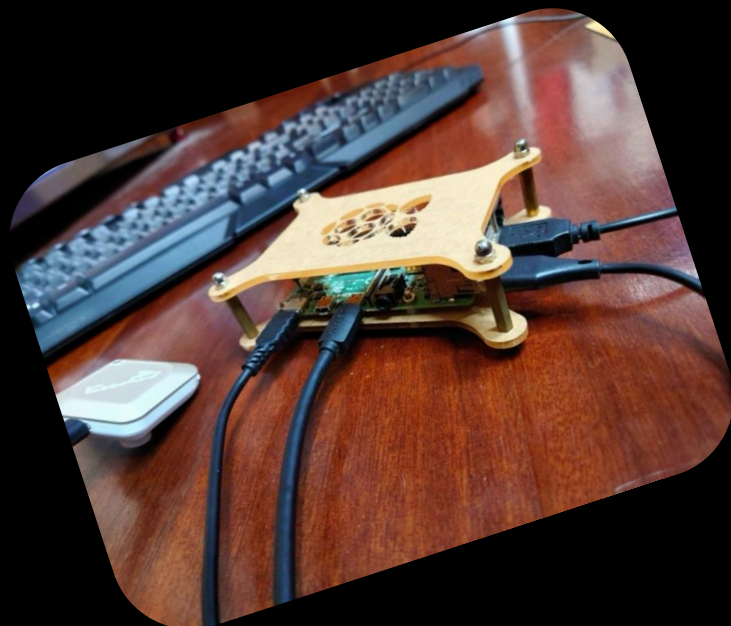
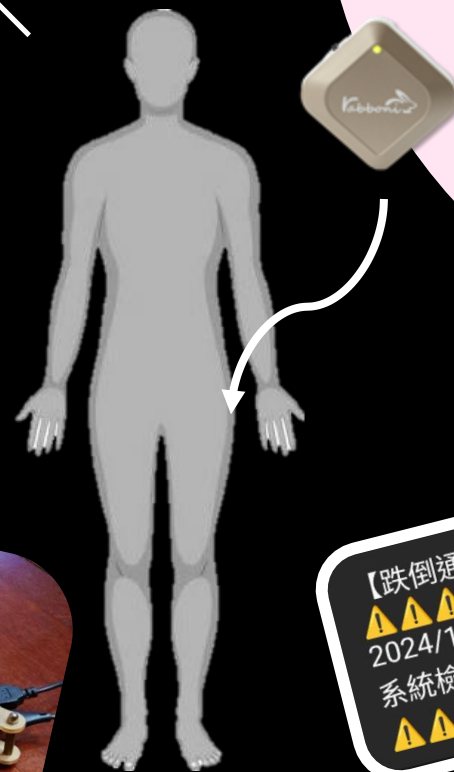
## 112年事故傷害人數

▽資料來源:衛生福利部112年國人死因統計結果



# 關於跌倒警示裝置

這是一款設計給獨居老人的裝置，他放置於腰部，可感測使用者是否跌倒，並於跌倒實發送通知到Line，並通知家人或監護人，並速詳細時間，未來希望可以新增更多的資訊，如定位等。



跌倒警示裝置



# 實作方式



## 材料

- Rabboni
- 樹莓派
- 輔助用電腦
- Python
- Line Notify

## 實際研究

起初我們希望可以透過感測 X Y Z軸中同時有兩個軸有移動即算跌倒，但經過實測後發現他會對走路發生誤判的情形，因此我們改成Y軸有下落且X或Z軸有大活動裁判定為跌倒，最後將數值微調並加入Line通知功能變完成了此項作品。



# 完整程式

```
import struct
from bluepy.btle import *
import binascii
import threading
import requests
import time
import math

# 用於儲存加速度數據
savedatas = []

# 判斷是否發生跌倒
def is_fall(acc_x, acc_y, acc_z, rabboni_x, rabboni_y, rabboni_z):
    # 結合 Y 軸和其他軸條件, 判斷是否跌倒
    if acc_y < 0.6 and (abs(acc_x) > 0.7 or abs(acc_z) > 0.7):
        return True
    return False

# 將十六進位數據轉為十進位, 並處理二補數
def com(temp):
    dec = int(temp, 16)
    if dec > 32768:
        dec = ((dec ^ 0xFFFF) + 1) * -1
    dec = dec * 2 / 32768
    return dec

# 接收資料
class MyDelegate(DefaultDelegate):
```

# 完整程式



```
# 接收資料
class MyDelegate(DefaultDelegate):
    def __init__(self, data):
        DefaultDelegate.__init__(self)
        self.data = data

    def handleNotification(self, cHandle, data):
        # 將接收到的數據轉為十六進位
        t1 = binascii.hexlify(data)[0:4]
        t2 = binascii.hexlify(data)[4:8]
        t3 = binascii.hexlify(data)[8:12]
        t4 = binascii.hexlify(data)[12:16]
        t5 = binascii.hexlify(data)[16:20]
        t6 = binascii.hexlify(data)[20:24]

        # 將數據轉為加速度和角速度值
        acc_x = com(t1)
        acc_y = com(t2)
        acc_z = com(t3)

        rabboni_x = com(t4) * 500
        rabboni_y = com(t5) * 500
        rabboni_z = com(t6) * 500

        # 將加速度數據加入全域列表
        savedatas.append([acc_x, acc_y, acc_z])
```



# 完整程式



```
# 將加速度數據加入全域列表
savedatas.append([acc_x, acc_y, acc_z])

# 判斷是否跌倒
did = is_fall(acc_x, acc_y, acc_z, rabboni_x, rabboni_y,
rabboni_z)

if did:
    # 記錄時間並發送通知
    t = time.time()
    tt = time.localtime(t)
    print("fall down !!!")
    t = time.time()
    t1 = time.localtime(t)
    t2 = time.strftime('%Y/%m/%d %H:%M:%S', t1)
    url = 'https://notify-api.line.me/api/notify'
    token = '*****'
    headers = {
        'Authorization': 'Bearer ' + token    # 設定權杖
    }
    data = {
        'message': '\n🚧🚧🚧🚧🚧🚧🚧🚧🚧🚧\n' +
time.strftime('%Y/%m/%d %H:%M:%S', tt) + '\n系統檢測到跌倒，請立即採取行
動!\n🚧🚧🚧🚧🚧🚧🚧🚧🚧🚧 '    # 設定要發送的訊息
    }
    data = requests.post(url, headers=headers, data=data)    #
使用 POST 方法
```

# 完整程式

```
        data = requests.post(url, headers=headers, data=data) #
使用 POST 方法

        # 將加速度值加入清單
        self.data.append((acc_x, acc_y, acc_z))

# 掃描藍牙設備
scanner = Scanner()
devices = scanner.scan(10.0)

print("finding device...")

# 指定設備地址
address = "C4:F2:20:B4:25:78"

# 連接到指定設備
per = Peripheral(address, "random")

print('\n-----start-----\n-
\n-
\n-
\n-----start-----')
try:
    # 獲取設備所有服務
    services_dic = per.getServices()
    for service in services_dic:
        charac_dic = service.getCharacteristics()

# 創建列表以儲存數據
data = []
```

# 完整程式



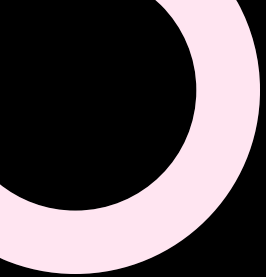
```
# 創建列表以儲存數據
data = []

# 設定委派處理通知
per.setDelegate(MyDelegate(data))

# 啟用通知
setup_data = b"\x01\x00" # 啟用通知的數據
led_data = b"\x37\x01\x01" # LED 控制數據

# 寫入特徵值以開啟設備功能
per.writeCharacteristic(45, led_data, withResponse=True)
per.writeCharacteristic(23, setup_data, withResponse=True)

# 等待並處理通知
while True:
    if per.waitForNotifications(5.0):
        continue
finally:
    # 斷開連接並儲存數據到文件
    per.disconnect()
    with open('data.json', "w") as f:
        f.write(str(savedatas))
```

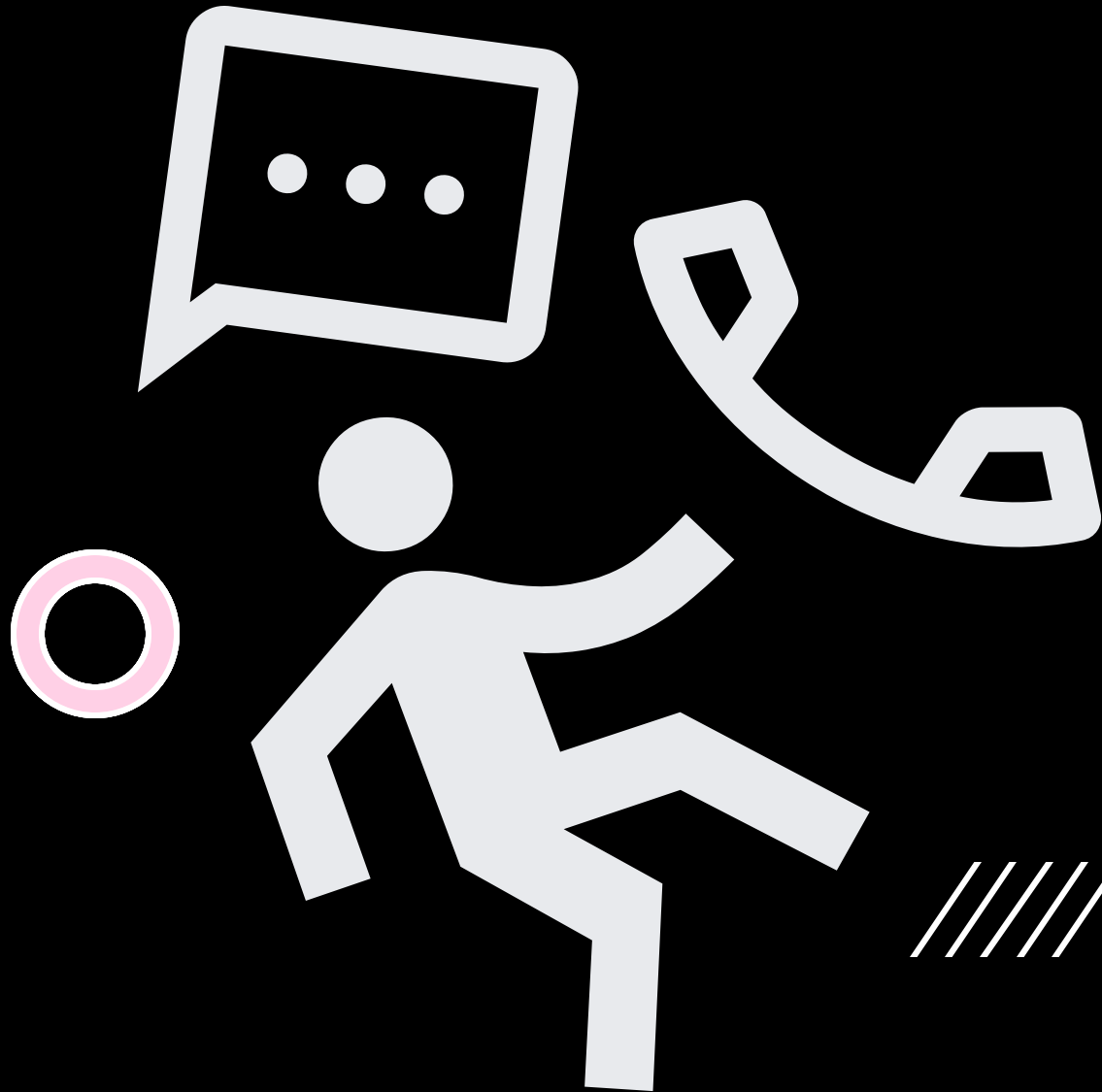


# 實際展示



# 未來展望

未來，我們希望可以進一步優化這款裝置，將其體積縮小以提升便攜性，同時提高系統的靈敏度與準確性與發展定位功能。我們還希望能夠整合更多的平台，如簡訊、通話，以避免憾事再一次發生。



# 感謝您的聆聽

從哪裡跌倒，就從哪裡爬起來。  
*Pick yourself up from where you've fallen.*

