KaiKuTeK

KKT Module user guide

2022/05/17

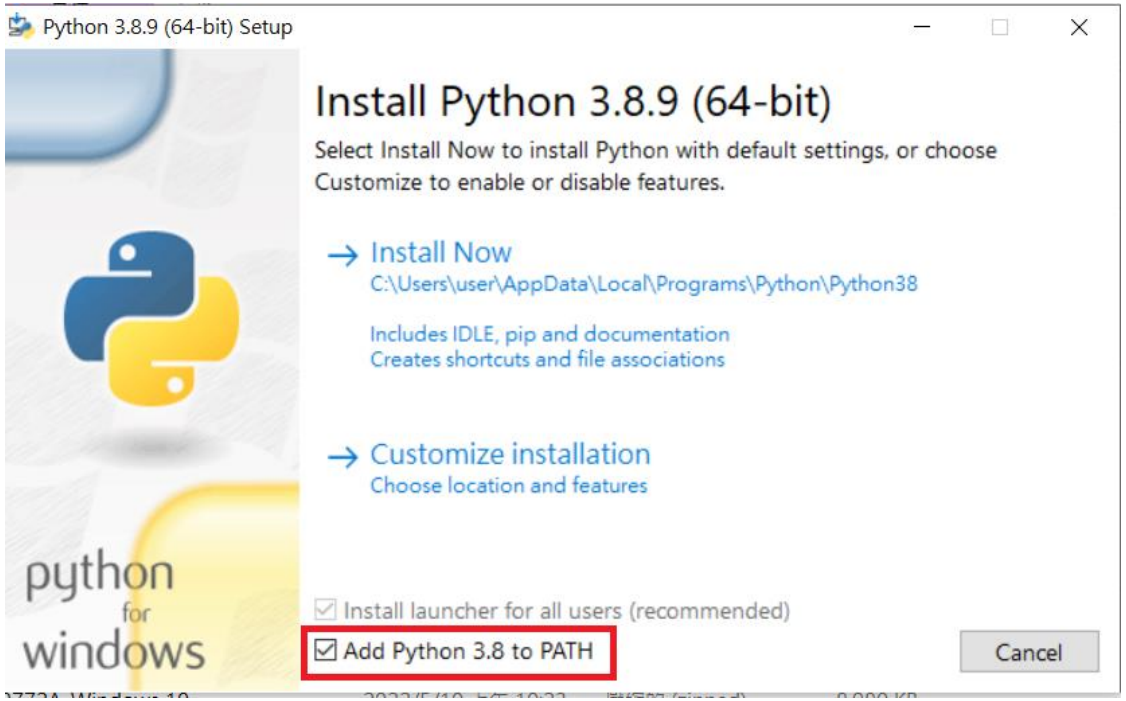
Echoing the Future – Bridging the World

Contents

- 安裝環境
- 拿取數據

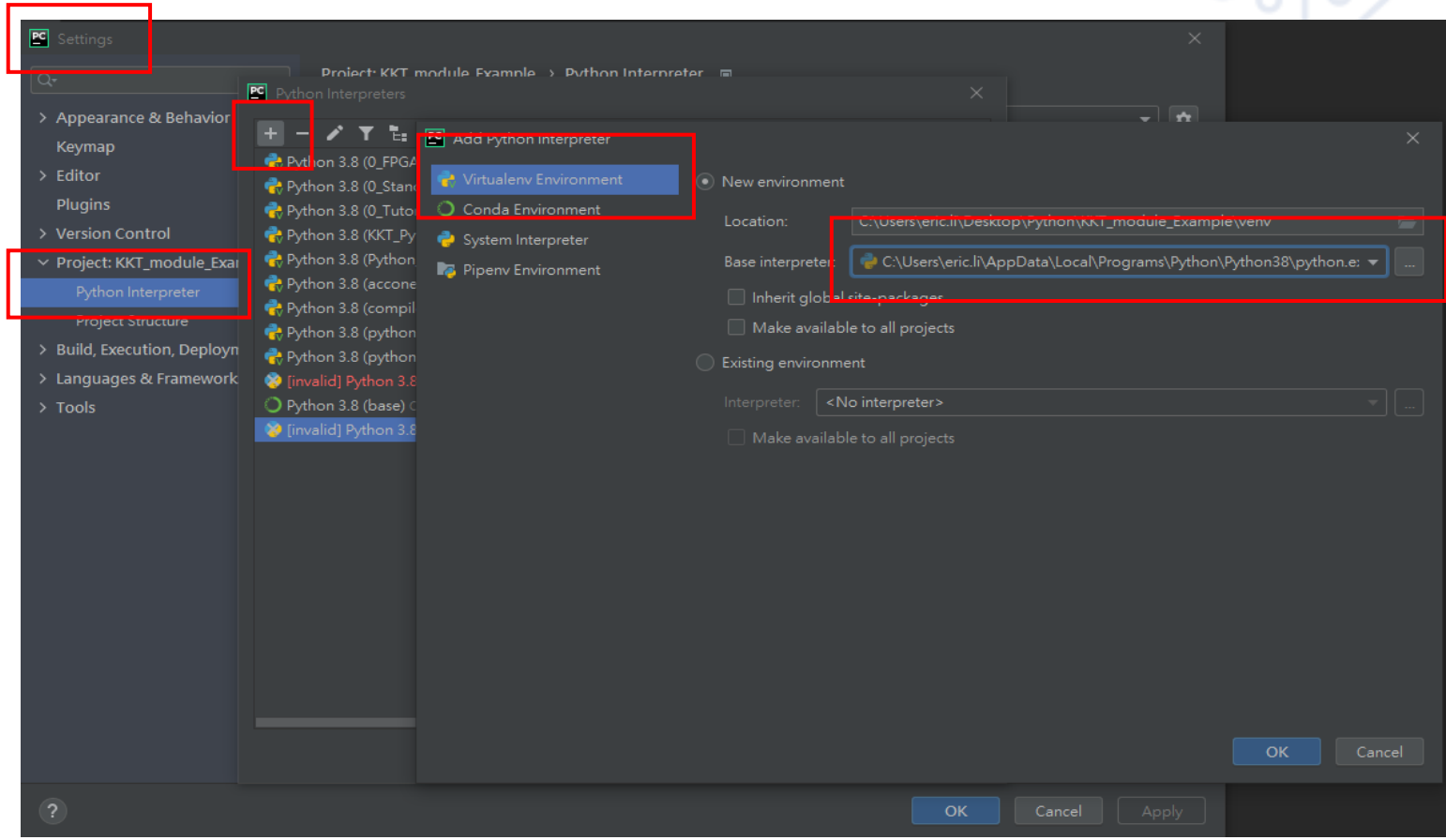
安裝環境(1/3)

- 安裝python：請使用 python 3.8.9 x64 版本
  - 安裝檔路徑：KKT_Module_Example\說明\python-3.8.9-amd64.exe
- 安裝pycharm：
  - 安裝檔路徑： KKT_Module_Example\說明\pycharm-community-2021.1.1.exe



安裝環境(2/3)

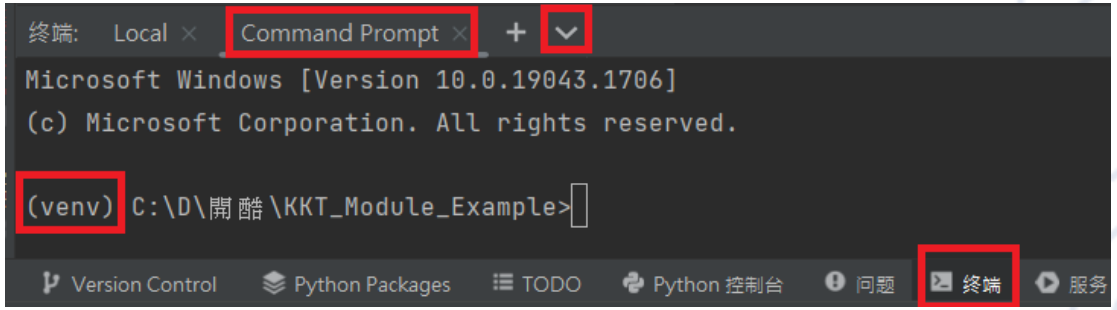
- 使用pycharm創建環境
  - 安裝完 pycharm 後選擇開啟專案 (KKT_Module_Example), 開啟專案時的設定可先用預設，進入後會再設定
  - 選擇完專案後點選左上角 File -> setting -> Project -> python Interpreter -> python Interpreter列表 -> show all -> 點擊加號新增一個編譯環境 -> 創建新virtual environment -> 選擇 python 版本(3.8)



- 提醒：建議每個專案建立各別的編譯環境, 且建議先建立乾淨的環境後再安裝所需套件, 以確保不會有環境不同以及模組版本衝突的問題

安裝環境(3/3)

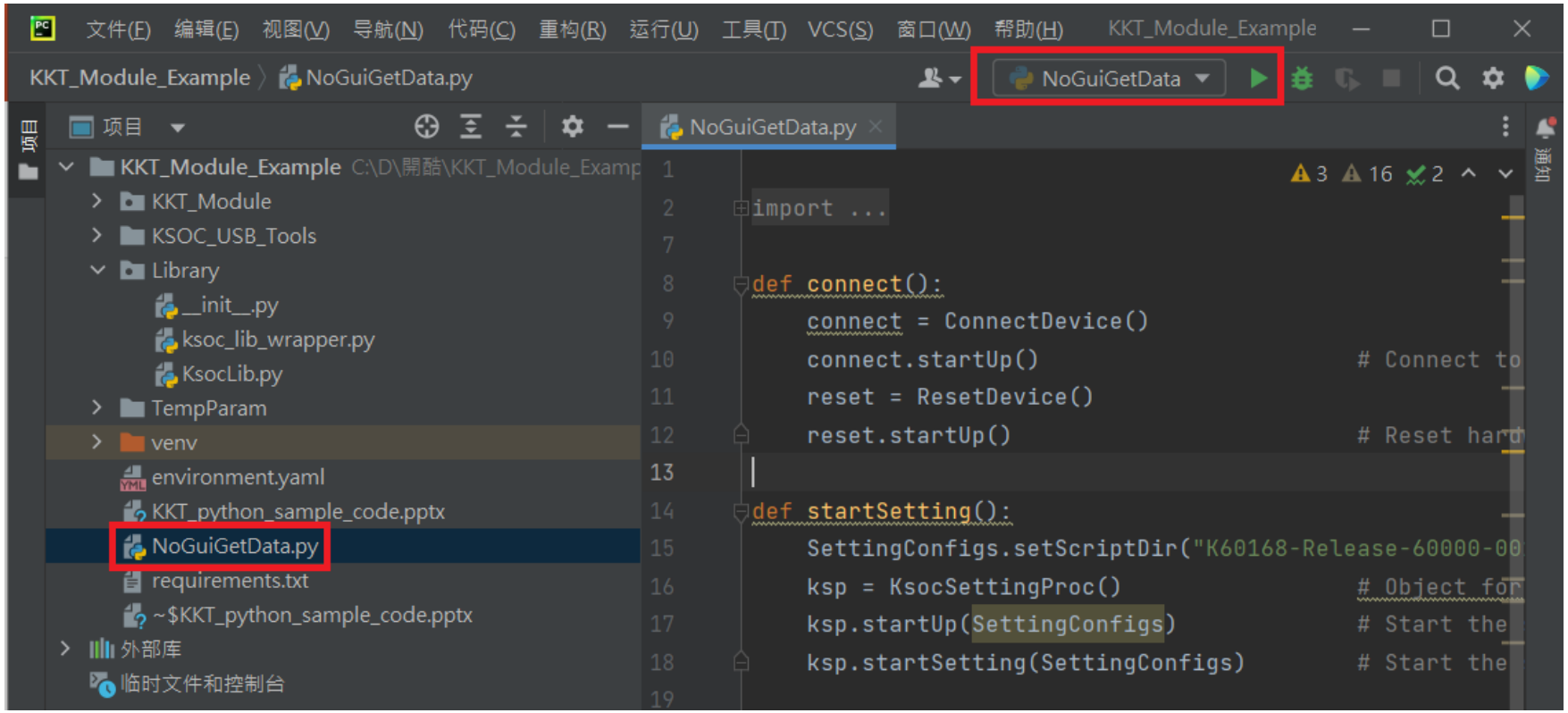
- 使用 pycharm 創建環境
  - 檢查Terminal頁籤的Command Prompt，若無可於右圖中紅框向下箭頭處點擊,
  - 查看命令前方是否為設定之環境(可能需要關掉重開Command Prompt才會更新)
  - requirements.txt文件編輯：torch相關的三個套件無法直接安裝，先將這三行刪除後再安裝，因之後從專案中導出的requirements.txt都會有這三個，故之後此處皆須注意
  - 使用requirements.txt 安裝套件：於Command Prompt中下指令安裝套件,
    指令：pip install -r requirements.txt
  - 環境安裝完成




```
threadpoolctl==3.0.0
tifffile==2022.4.8
torch==1.7.1+cpu
torchaudio==0.7.2
torchvision==0.8.2+cpu
typing-extensions==3.10.0.2
urllib3==1.26.7
websocket-client==1.2.1
wincertstore==0.2
```

拿取數據(1/6)

- 連接電路板後執行範例 NoGuiGetData.py
- 於專案中開啟NoGuiGetData.py，點擊運行後執行



拿取數據(2/6)

- 程式流程
  - 連接裝置 - connect()
  - 設定Parameter – startSetting()
  - 獲取手勢編號：startLoop()

```
def main():
    kgl.setLib()

    kgl.ksoclib.switchLogMode(False)

    connect()                              # First you have to connect to the device

    startSetting()                        # Second you have to set the setting configs

    startLoop()                            # Last you can continue to get the data in the loop
```

拿取數據(3/6)

- connect()流程
  - 確認裝置是否連接，以及顯示連接的方式
  - 重置裝置的hardware register

```
def connect():
    connect = ConnectDevice()
    connect.startUp()                       # Connect to the device
    reset = ResetDevice()
    reset.startUp()                         # Reset hardware register

def startSetting():...

def startLoop():
    # R = RawDataReceiver(chirps=32)          # Receiver for getting Raw data
    # R = FeatureMapReceiver(chirps=32)       # Receiver for getting RDI PHD map
    R = GesturesReceiver()                  # Receiver for getting hardware results (gestures, Axes, exponential)
    buffer = DataBuffer(100)                # Buffer for saving latest frames of data
    R.trigger()                             # Trigger receiver before getting the data
    time.sleep(0.5)
    print('# ======== Start getting gesture ===========')
    while True:                             # loop for getting the data
        res = R.getResults()                # Get data from receiver
        if res is None:
            continue
        print('Gesture = {}'.format(res[1]))                        # Print results
        '''
        Application for the data.
        '''
```

拿取數據(4/6)

- startSetting()流程
  - 輸入參數檔檔名(路徑：KKT_Module_Example\TempParam)
  - 根據檔案設定參數

```
def startSetting():                                    參數檔資料夾名稱
    SettingConfigs.setScriptDir("K60168-Release-60000-001-v1.3.0-rc0-20210624 - RX23")  # Set the setting folder name
    ksp = KsocSettingProc()                 # Object for setting process to setup the Hardware AI and RF before receive data
    ksp.startUp(SettingConfigs)             # Start the setting process
    # ksp.startSetting(SettingConfigs)        # Start the setting process in sub_thread
```

拿取數據(5/6)

- startLoop()流程
  - 在迴圈中會不斷的拿取裝置上的資料
  - 預設先抓10個frame的資料

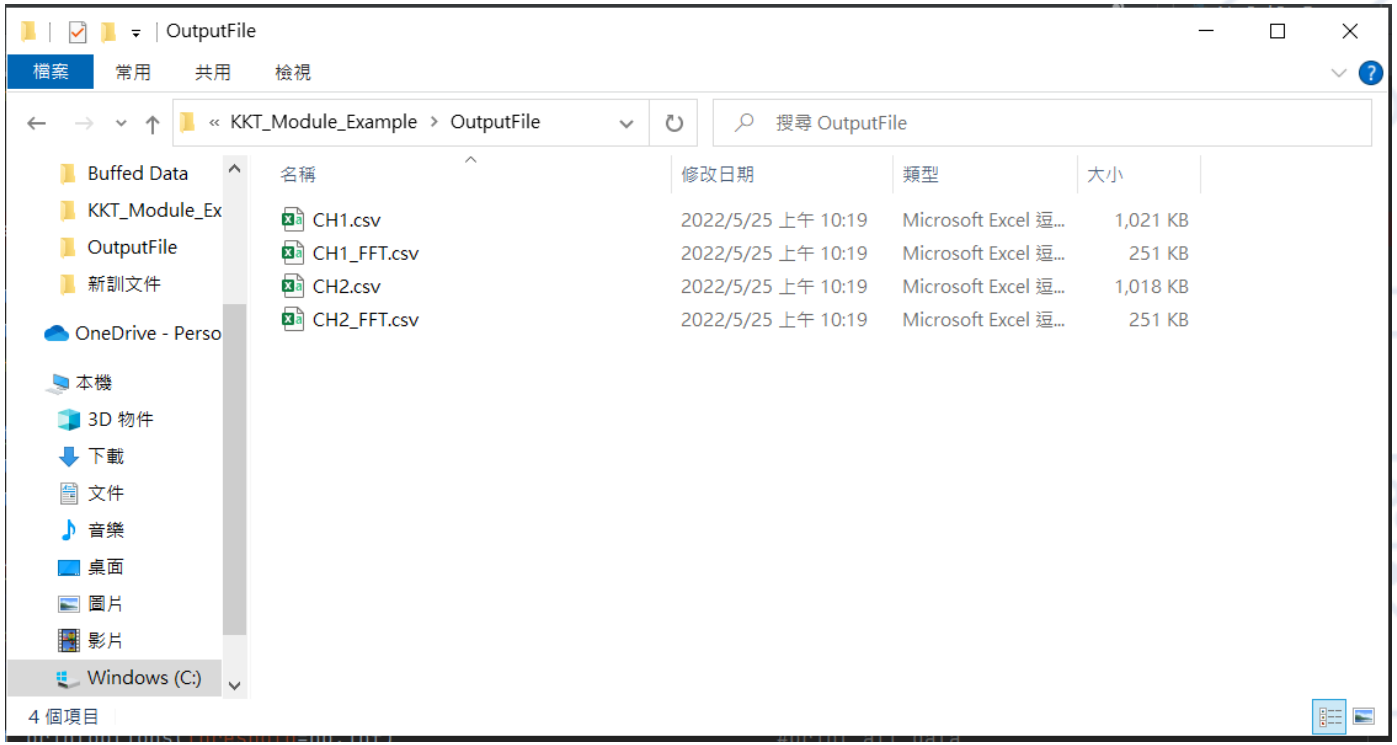
```
def startLoop():
    R = RawDataReceiver(chirps=32)          # Receiver for getting Raw data
    # R = FeatureMapReceiver(chirps=32)       # Receiver for getting RDI PHD map
    #R = GesturesReceiver()                  # Receiver for getting hardware results (gestures, Axes, exponential)
    buffer = DataBuffer(100)                # Buffer for saving latest frames of data
    R.trigger()                             # Trigger receiver before getting the data
    time.sleep(0.5)
    print('# ======== Start getting gesture ============')
    DirPath = os.path.normpath(os.path.join(os.getcwd(), '.\OutputFile'))
    if os.path.isdir(DirPath):
        shutil.rmtree(DirPath)
        os.mkdir(DirPath)
    else:
        os.mkdir(DirPath)

    # frameCount = input("Please input the number of frame you want to get：")
    frameCount = 10 #int(frameCount)   預設先抓10個frame，如需調整請改變此參數
    whileCount = 0
    all_res_ch1 = np.array([])
    all_res_ch2 = np.array([])
    all_fft_ch1 = np.array([])
    all_fft_ch2 = np.array([])
    while whileCount < frameCount:                          # loop for getting the data
        if keyboard.is_pressed('Esc'):
            print("\nyou pressed Esc, so exiting...")
            sys.exit(0)

        res = R.getResults()                # Get data from receiver
```

拿取數據(6/6)

- 匯出CSV檔路徑：KKT_Module_Example\OutputFile
- 每次匯出會將舊檔案刪掉，如需保存請先搬移到別處
- 檔案分別為：
  - CH1.csv：CH1的RawData
  - CH1_FFT.csv：CH1經傅立葉轉換後的資料
  - CH2.csv：CH2的RawData
  - CH2_FFT.csv：CH2經傅立葉轉換後的資料




KaiKuTeK
Thank You!
www.kaikutek.com