



Powering Innovation That Drives Human Advancement

---

# 使用 PyEDB 加速 SI/PI模擬與分析工作坊

Lin, Ming Chih

# 課程概要

- **PyEDB** 是一個 Python 模組它是 **PyANSYS**的一部分專門用於與 **ANSYS Electronics Desktop** 的 **EDB**（**Electronics DataBase**）交互。**PyEDB** 提供了許多功能使用者可以在 Python 環境中直接操作和分析 **PCB**（印刷電路板）和封裝設計的資料。
- 透過 **PyEDB**使用者可以：
  - 讀取和修改 **PCB** 或封裝的結構和屬性。
  - 提取電氣元件的屬性如阻抗、連接點等。
  - 創建和修改層、網和元件等。
  - 執行信號完整性和電源完整性分析。
- 工作坊將從 **PyEDB** 的基本概念開始解釋其與 **PyAEDT** 的關係以及如何設置和開始使用 **PyEDB**。隨後我們將介紹一些實用的功能包括讀取 **PCB** 設計資料、修改元件屬性和層結構以及進行信號完整性和電源完整性分析的基本方法。

# PyEDB 對 SI/PI 工程師的優勢





- 設計修改與模擬條件設定
  - 快速修改材料屬性和 **PCB** 堆疊結構。
  - 靈活設定激發源、端口配置和模擬邊界條件。
- 無 GUI 高效操作
  - 無需圖形使用者界面通過腳本直接操作提升處理速度。
  - 在記憶體中高效處理大規模和複雜設計適合大量仿真需求。
- 代碼與設計復用
  - 支持代碼重用降低重複工作提升開發效率。
  - 建立模塊化設計元件庫方便在不同項目中重用確保設計質量。

# PyEDB 官方範例

[https://edb.docs.pyansys.com/version/stable/examples/legacy\\_pyaedt\\_integration/index.html](https://edb.docs.pyansys.com/version/stable/examples/legacy_pyaedt_integration/index.html)

**PyAnsys**

Getting startedUser guideAPI reference**Examples**Contribute


Search  ctrl + k0.30 (stable)    

**Section Navigation**

**AEDT integration**

EDB: 5G linear array antennaEDB: Layout Components**EDB: fully parametrized design**EDB: Pin to Pin projectEDB: geometry creationEDB: Layout Creation and SetupEDB: parameterized designEDB: Network Analysis in Siwave

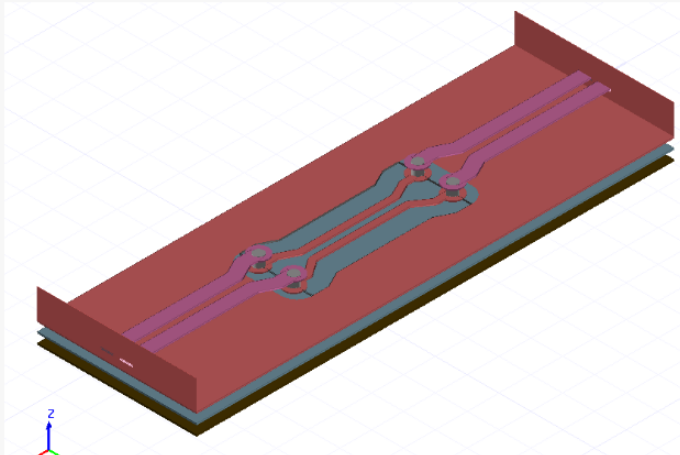
StandaloneUse configuration


PyAnsys >  > Examples > AEDT integration > EDB: fully parametrized design

## EDB: fully parametrized design

This example shows how to use the EDB interface along with HFSS 3D Layout to create and solve a parameterized layout. The layout shows a differential via transition on a printed circuit board with back-to-back microstrip to stripline transitions. The model is fully parameterized to enable investigation of the transition performance on the many degrees of freedom.

The resulting model is shown below




 On this page


Set non-graphical mode

Launch EDB.

Draw parametrized traces

Add HFSS simulation setup

 Edit on GitHub

 Show Source

# Agenda

- Part1
  - EDB介紹與增刪改查操作
  - 練習
- Part2
  - SI/PI模擬設定介紹
  - 練習
- 連結PyAEDT進行模擬並輸出結果



# Part1: EDB介紹與增刪改查操作

# AEDB 文件格式

- 高效的數據管理

- AEDB 文件格式專為高效管理複雜的電子設計數據而設計支持快速訪問和處理。

- 專為 **Ansys** 應用優化

- 直接支持於 **Ansys Electronics Desktop** 和相關的電磁模擬軟件如 **HFSS**、**SIwave** 允許用戶直接從這些工具中導入、修改和運行仿真。

- 支持多種工作流程

- 支持包括信號完整性（SI）、功率完整性（PI-DC）和電熱分析在內的多種工作流程。

- 便於協作與共享

- 促進跨團隊和工具的協作便於設計資源的共享和重用提高設計一致性和質量。

# aedb目錄的組成

- 從圖片中可以看到.aedb目錄包括以下兩個主要文件或文件夾：
  - **stride** 文件夾 - 這個文件夾包含與特定設計的步進數據或版本有關的信息。通常這樣的文件夾會用於組織和存儲過程中的臨時數據或具有不同設計階段的數據。
  - **edb.def** 文件 - 包含定義電子數據庫中特定部分的設定信息。這類文件一般用於保存設計的定義參數使得用戶能夠按照既定的規範重用或

	stride	修改日期: 2024/8/8 上午 10:00
	edb.def 類型: Export Definition File	修改日期: 2024/8/8 上午 09:47 大小: 2.26 MB



# EDB資料庫當中包含資料

- 這個資料庫專門存儲關於印刷電路板（**PCB**）或集成電路（**IC**）設計的各種數據。
  - 材料（**Materials**）存儲有關用於製造電路板的各種材料的資訊例如導電材料、絕緣材料等。
  - 堆疊結構（**Stackup**）：描述**PCB**的層疊結構包括各層的材料、厚度和其他物理特性。
  - 網絡（**Nets**）：表示電路板上的電氣連接。一個網絡包括連接一組電子元件的導體。
  - 層（**Layers**）：描述**PCB**中的各個層次包含信號層、地層或電源層等不同的功能層。
  - 墊片（**Padstacks**）：描述孔的結構這些孔用於放置連接元件引腳的通孔或埋孔。
  - 引腳（**Pins**）：元件引腳的資料通常是與特定的墊片相關聯用於建立元件。
  - 元件（**Components**）：涉及的所有電子元件資訊包括它們的規格、位置和方向。
  - 激勵（**Excitations**）：用於模擬中的電源激勵設置用於性能分析。
  - 設定（**Setups**）：包含模擬設定如頻率範圍、解析度等這些設定確定了模擬執行的具體條件。

# PyEDB 的優勢與功能

- **Python 交互性**
  - PyEDB 允許使用者直接通過 Python 與 Ansys Electronics Database (EDB) 進行交互使程式設計更為直觀和靈活。
- **功能完整**
  - 提供全面的 API 支持涵蓋從基本數據操作到高級仿真設置的所有功能。
- **速度快捷**
  - 在記憶體中運行無需用戶界面極大提高了操作大型和複雜佈局設計的速度與效率。
- **靈活的運行環境：**
  - 適用於 Windows 和 Linux 支持圖形化與非圖形化工作流程。
- **自動化與效率：**
  - 支持批處理和集群作業調度適合高效率的電子設計自動化。

# 從其他檔案格式如.brd .gdsII .odb++等轉換成.aedb

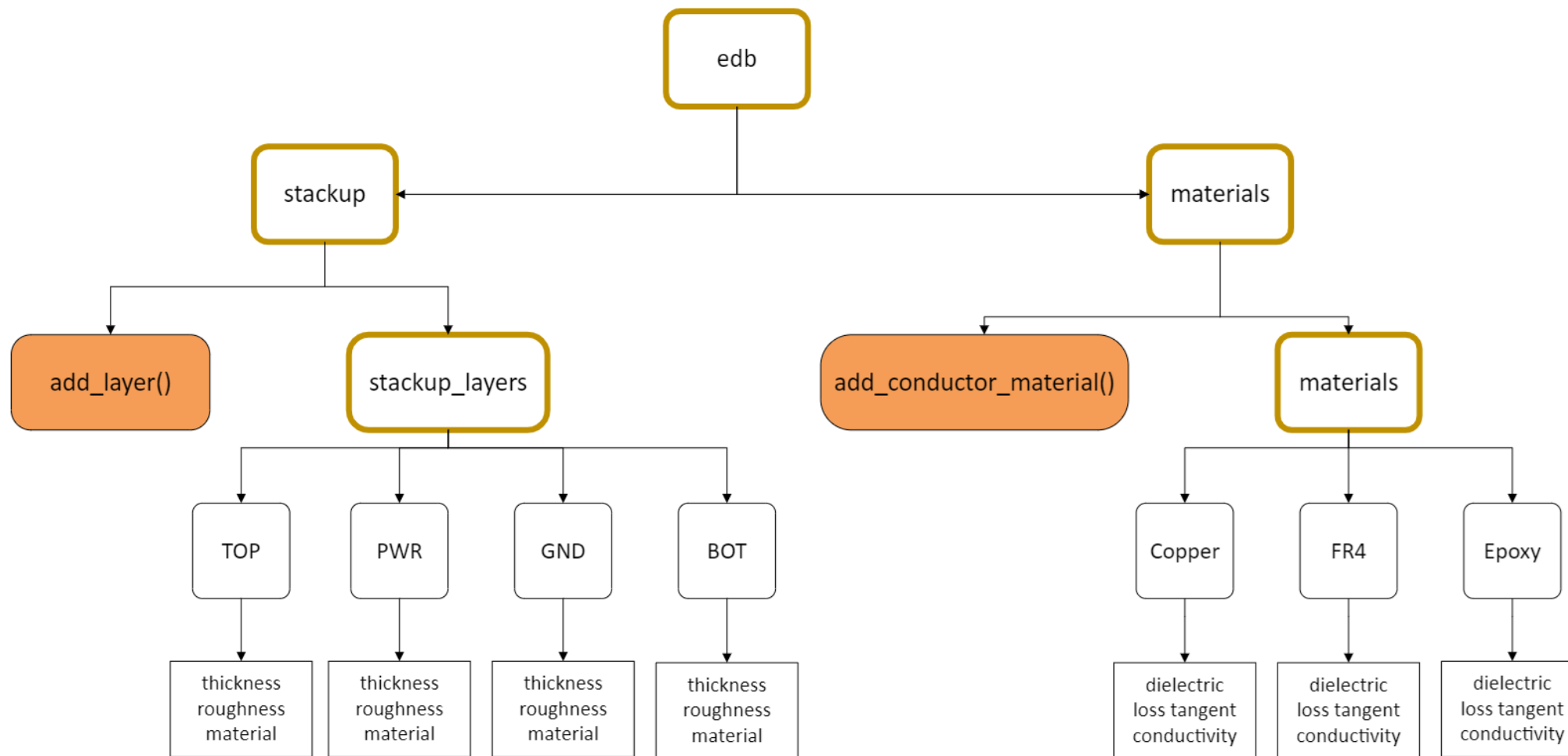
- 要從其他文件格式如 .brd（電路板設計文件）.gdsII（積體電路設計）或 .odb++（PCB 和封裝設計的數據交換格式）轉換成 .aedb（Ansys Electronics DataBase）您可以通過以下幾種方法：
  - 使用 Ansys Electronics Desktop (AEDT)：
    - AEDT提供了導入工具來直接將這些文件類型導入到 AEDT 中然後可以將導入的設計保存為 .aedb 格式。在 AEDT 中打開應用程序選擇相應的導入選項載入您的 .brd 或 .odb++ 文件並按照提示進行操作來完成導入和保存過程。
  - 使用PyEDB：
    - PyEDB 允許用戶直接從 Python 環境中訪問和操作 Ansys Electronics Database（EDB）。它不僅提供了操作 EDB 的接口還支援將多種常見的電子設計文件格式轉換為 .aedb 格式。這些格式包括 PCB 設計文件的 BRD、用於 PCB 和組件製造的 IPC2581 XML、積體電路設計的 GDSII以及 CAD 繪圖的 DXF 文件。

# PyEDB如何連結資料庫

- 這行Python程式碼從pyedb模組匯入Edb物件並建立了一個 Edb 物件的實例指向一個位於 d:/demo/目錄下的 .aedb 文件。
- edbversion=2024.1 指定了文件的版本這是必須與你的文件兼容的版本。
- 一旦數據庫文件被載入你就可以使用 edb 物件來進行各種操作如查詢、更新或處理數據。具體的操作會依賴於 Edb 類提供哪些方法和功能。

```
From pyedb import Edb  
edb = Edb(d:/demo/pcb.aedb edbversion=2024.1)
```

# Edb資料庫的架構圖(簡略)



# 架構圖說明

- 架構圖當中粗框的物件（如**edb**、**stackup**、**materials**）代表的是唯一的物件它們負責管理或包含容器的物件（如**stackup\_layers**、**materials**）。透過某些函數（如**add\_layer()**或**add\_conductor\_material()**）可以將新的物件加入到這些容器中的。以下是這些容器與它們包含的物件之間的關係：
  - **edb** :
    - 這是整個電子設計佈局（**EDB**）的主體管理設計中的所有相關資料。
  - **stackup** :
    - 堆疊層的管理物件底下有**stackup\_layers**容器。你可以使用**add\_layer()**函數將新的層（如**TOP**、**PWR**、**GND**、**BOT**）加入到**stackup\_layers**這個容器中。每一層都有屬性如厚度、粗糙度、材料等。
  - **materials** :
    - 管理所有材料的唯一管理物件。你可以透過**add\_conductor\_material()**函數將導體材料（如**Copper**）或介電材料（如**FR4**、**Epoxy**）加入到底下**materials**容器中。這些材料具有諸如介電損耗正切、導電性等屬性。
- 其他像是**edb.modeler** **edb.padstacks** **edb.components**都是以類似方式運作

# 查詢資料庫所有材料

- 使用 `print(edb.materials.materials)` 這樣的命令時會看到一個包含多個材料物件的字典格式輸出。每個材料都被表示為一個物件這裡解釋輸出的各部分：
  - 字典的鍵（**Key**）：每個鍵是一個字符串代表材料的名稱例如 **AIR**、**COPPER**、**FR-4** 等
  - 字典的值（**Value**）：每個值是一個 **Material** 物件。這些物件存儲了關於材料的具體屬性如其電學、磁學和熱學性質。

```
In [14]: print(edb.materials.materials)
{'AIR': <pyedb.dotnet.edb_core.materials.Material object at 0x000002308A8E2C80>,
 'BOTTOM_FILL': <pyedb.dotnet.edb_core.materials.Material object at 0x000002308A8E2D10>,
 'COPPER': <pyedb.dotnet.edb_core.materials.Material object at 0x000002308A8E27A0>,
 'FR-4': <pyedb.dotnet.edb_core.materials.Material object at 0x000002308A8E28F0>,
 'FR-4_1': <pyedb.dotnet.edb_core.materials.Material object at 0x000002308A8E2950>,
 'FR-4_2': <pyedb.dotnet.edb_core.materials.Material object at 0x000002308A8E2890>,
 'FR4_epoxy': <pyedb.dotnet.edb_core.materials.Material object at 0x000002308A8E28C0>,
 'GND_FILL': <pyedb.dotnet.edb_core.materials.Material object at 0x000002308A8E2980>,
 'LYR_1_FILL': <pyedb.dotnet.edb_core.materials.Material object at 0x000002308A8E2800>,
 'LYR_2_FILL': <pyedb.dotnet.edb_core.materials.Material object at 0x000002308A8E2830>,
 'PWR_FILL': <pyedb.dotnet.edb_core.materials.Material object at 0x000002308A8E27D0>,
 'TOP_FILL': <pyedb.dotnet.edb_core.materials.Material object at 0x000002308A8E2860>}
```



# 查詢特定材料底下有哪些屬性

- 當使用 `print(dir(edb.materials.materials[FR-4]))` 命令時會獲得材料對象 FR-4 所具有的所有屬性和方法的列表。包括了公開和私有屬性以及方法以下是其中部分關鍵屬性和方法的簡要說明：
  - 私有屬性（以兩個底線開頭如 `__init__`）：這些是對象的內部屬性和方法通常不用於直接調用但是它們對於了解物件如何管理其數據和行為是有幫助的。
  - 公開屬性（沒有底線前綴）：`name`：材料的名稱。`conductivity`：導電率。`permittivity`：介電常數。`permeability`：磁導率。`mass_density`：質量密度。`thermal_conductivity`：熱導率。`specific_heat`：比熱。`thermal_expansion_coefficient`：熱膨脹系數。`youngs_modulus`：楊氏模量。`poisson_ratio`：泊松比。

```
In [17]: print(dir(edb.materials.materials['FR-4']))
['_Material__dc_model', '_Material__edb', '_Material__edb_definition',
'_Material__edb_value', '_Material__load_all_properties', '_Material__material_def',
'_Material__name', '_Material__properties', '_Material__property_value', '__class__',
'__delattr__', '__dict__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__',
'__getattr__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__le__',
'__lt__', '__module__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__',
'__setattr__', '__sizeof__', '__str__', '__subclasshook__', '__weakref__',
'conductivity', 'dc_conductivity', 'dc_model', 'dc_permittivity',
'dielectric_loss_tangent', 'dielectric_model_frequency', 'loss_tangent',
'loss_tangent_at_frequency', 'magnetic_loss_tangent', 'mass_density', 'name',
'permeability', 'permittivity', 'permittivity_at_frequency', 'poisson_ratio',
'specific_heat', 'thermal_conductivity', 'thermal_expansion_coefficient', 'to_dict',
'update', 'youngs_modulus']
```



# 查詢及修改特定材料屬性值

- 我們可以看到對 **FR-4** 材料的介電常數進行了查詢和修改的操作：
  - 查詢介電常數：初始查詢顯示 **FR-4** 材料的介電常數為 **3.86**。
  - 修改介電常數：將 **FR-4** 材料的介電常數修改為 **4.2**。
  - 重新查詢介電常數：修改後再次查詢顯示新的介電常數值為 **4.2**。
- 這表明成功地使用 **PyEDB** 更新了資料庫當中材料的物理性質。這種能力對於調整和優化電子設計中的特性至關重要尤其是當需要根據不同的設計要求或環境條件進行調整時。

```
In [18]: print(edb.materials.materials['FR-4'].permittivity)
3.86

In [19]: edb.materials.materials['FR-4'].permittivity = 4.2

In [20]: print(edb.materials.materials['FR-4'].permittivity)
4.2
```

# 保存對EDB文件的修改

- 當你想要保存對EDB (Electronic Design Database) 文件的修改或將其另存為新的文件時使用如 `save_edb_as()` 方法這樣的函數是一種常見的做法。根據你提供的代碼這裡有一個如何使用該方法來保存文件的步驟說明：在你的 `edb` 物件上調用 `save_edb_as()` 方法並傳入新文件的路徑作為參數。

```
In [21]: edb.save_as('d:/demo/pcb_new.aedb')
```

```
In [22]:
```

# Lab 1. 以阻值分類所有電阻

- 學習重點:
  - 判斷用何種資料結構儲存 list? tuple? dict? set?
  - 找出edb當中存放電阻的容器
  - 讀出電容ID及電容值並存放到資料結構
  - 輸出結果

## Lab 2. 找出V3P3\_S5上的去耦合電容

- 學習重點:
  - 找出位於V3P3\_S5的電容
  - 找出這些電容的另一支pin腳連接的Net
  - 如果電容是連接V3P3\_S5與GND則為去耦合電容
  - 輸出去耦合電容編號

## Lab 3. 建立新材料並設定疊構

- 學習重點:
  - 建立資料結構儲存屬性每一層dk df conductivity
  - 建立新edb
  - 在edb當中加入新的Layer並設定Layer屬性
  - 儲存edb
  - 開啟edb



# Part2: SI/PI模擬設定 並連結PyAEDT進行模擬並輸出結

# 使用 PyEDB 進行模擬設置和激勵配對

- 模擬前的 EDB 數據準備
  - 確保所有 EDB 資料準確無誤，包括材料、堆疊和佈局信息。
- 添加 Solver Setup 資訊
  - 在 EDB 中加入 Solver Setup 資訊，為模擬準備必要的配置參數。
- 根據模擬類型選擇激勵
  - HFSS：設定 ports 和必要的 geometric configurations，如 cutout 和 Air box。
  - SIwave DCIR：配置 source/sink 和 probes 來分析直流阻抗。
- 模擬環境設定
  - 配置模擬的具體條件，包括頻率範圍、電氣特性等，根據不同的 Solver 和激勵類型進行調整。

# edb

- create\_cutout
- create\_cutout\_multithread
- create\_cutout\_on\_point\_list
- cutout
- cutout\_multizone\_layout
- create\_hfss\_setup
- create\_hfsspi\_setup
- create\_raptorx\_setup
- create\_siwave\_dc\_setup
- create\_siwave\_syz\_setup
- hfss\_setups
- setups
- simsetupdata
- siwave\_ac\_setups
- siwave\_dc\_setups
- create\_port
- ports
- create\_current\_source
- create\_voltage\_source
- source
- source\_version
- sources
- create\_voltage\_probe probes



# edb.core\_hfss

- `configure_hfss_analysis_setup`
- `configure_hfss_extents`
- `create_bundle_wave_port`
- `create_circuit_port_on_net`
- `create_circuit_port_on_pin`
- `create_coax_port_on_component`
- `create_current_source_on_net`
- `create_current_source_on_pin`
- `create_differential_wave_port`
- `create_edge_port_horizontal`
- `create_edge_port_on_polygon`
- `create_edge_port_vertical`
- `create_hfss_ports_on_padstack`
- `create_lumped_port_on_net`
- `create_resistor_on_pin`
- `create_rlc_boundary_on_pins`
- `create_vertical_circuit_port_on_clipped_traces`
- `create_voltage_source_on_net`
- `create_voltage_source_on_pin`
- `create_wave_port`
- `excitations`
- `get_layout_bounding_box`
- `get_ports_number`
- `get_trace_width_for_traces_with_ports`
- `hfss_extent_info`
- `layout_defeaturing`
- `probes`
- `set_coax_port_attributes`
- `sources`
- `trim_component_reference_size`

# edb.core\_siwave

- add\_siwave\_dc\_analysis
- add\_siwave\_syz\_analysis
- configure\_siw\_analysis\_setup
- create\_circuit\_port\_on\_net
- create\_circuit\_port\_on\_pin
- create\_circuit\_port\_on\_pin\_group
- create\_current\_source\_on\_net
- create\_current\_source\_on\_pin
- create\_current\_source\_on\_pin\_group
- create\_dc\_terminal
- create\_exec\_file
- create\_impedance\_crosstalk\_scan
- create\_pin\_group
- create\_pin\_group\_on\_net
- create\_pin\_group\_terminal
- create\_port\_between\_pin\_and\_layer
- create\_resistor\_on\_pin
- create\_rlc\_component
- create\_voltage\_probe\_on\_pin\_group
- create\_voltage\_source\_on\_net
- create\_voltage\_source\_on\_pin
- create\_voltage\_source\_on\_pin\_group
- create\_vrm\_module
- excitations
- icepak\_component\_file
- icepak\_use\_minimal\_comp\_defaults
- pin\_groups
- place\_voltage\_probe
- probes
- sources
- voltage\_regulator\_modules

# PyEDB 與 PyAEDT 的整合應用內容：

- 將 PyEDB 載入到 AEDT 3D Layout
  - 完成 PyEDB 的建置後，使用 PyAEDT 將 PyEDB 數據導入 AEDT 3D Layout，便於視覺化和驗證設計設置的正確性。
- 模擬和觀察
  - 在 AEDT 3D Layout 中進行必要的模擬，觀察電磁行為和任何潛在問題，確保設計滿足預期的性能標準。
- 數據擷取與分析
  - 使用 PyAEDT 提取模擬結果，進行深入分析，識別關鍵的性能指標和潛在改進點。
- 生成報告
  - 結合 Python 強大數據能力，快速生成包含模擬結果和分析數據的專業報告，用於技術評估或項目審核。

# Lab 4. 建立差分對結構並設置WavePort

- 學習重點:
  - 讀取疊構檔案
  - 建立傳輸線與平面
  - 設wave port

# Lab 5. 設置 3D Layout SI 模擬

- 學習重點:
  - 導入EDB到PyAEDT
  - 執行模擬
  - 輸出結果

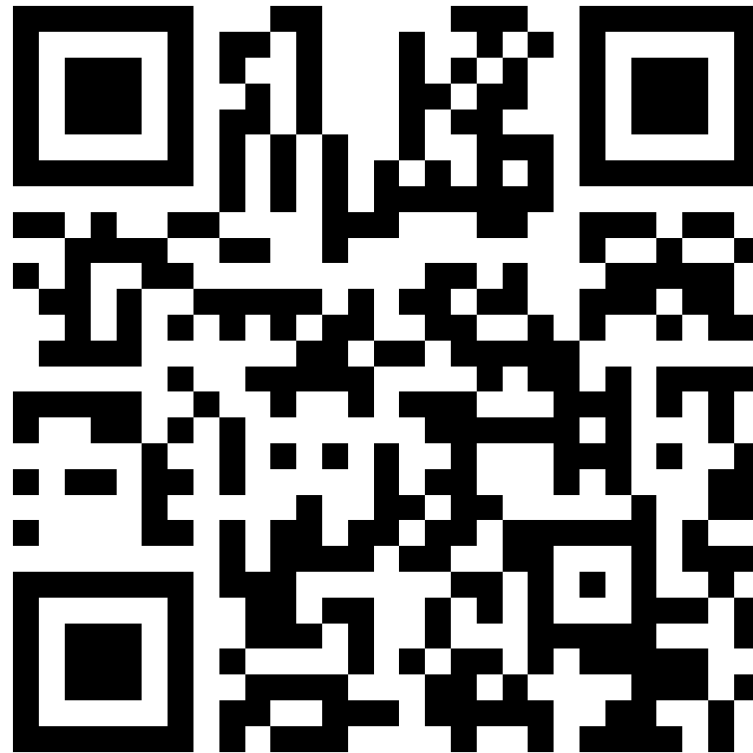
# Lab 6. 設置Siwave SYZ SI模擬

- 學習重點:
  - 設定ports
  - 設定Siwave模擬
  - 執行模擬並輸出S參數



請掃描QR Code填寫問卷！

如果有任何問題也歡迎您在  
問卷裡提出



填寫完問卷請記  
得找工作人員領  
取精美小禮喔！

<https://forms.office.com/r/KUfFD5E07P>



TAIWAN

# SIMULATION WORLD 2024

INSPIRE / EMPOWER / EQUIP

**Powering Innovation  
That Drives Human Advancement**

## 台灣用戶技術大會

聚焦 AI/ML 技術創新、半導體與先進封裝、電子系統分析與 IC 設計、電動車等熱門議題。

特別邀請到 Foxconn 鴻海資深協理、ITRI 副所長及 Ansys 副總裁、資深 R&D 總監及 Ansys/IEEE Fellow & Chief Technologist 等專家分享市場前沿技術主題演講。



活動圓滿落幕，  
歡迎觀看精彩內容





The Ansys logo, featuring a stylized yellow and black 'A' followed by the word 'nsys' in black.

