

● ECB 連線、設定

類別名稱	ECB
函式名稱	ECB_RunAsync
引數	Null
回傳	Task object
<p>此函式用於初始化 ECB 網路介面以及相關設定，並開啟新的執行序專門處理 ECB 網路封包。需先創造 ECB 物件後才能使用，並可搭配 IsConnected 屬性判斷 TCP 是否建立成功。</p> <p>Ex :</p> <pre>ECB ecb = new ECB{}; var ecbTask = ecb.RunAsync(); // Block in while loop if not connected. while (!ecb.IsConnected) { if(ecbTask.IsCompleted) { break; //連線失敗 } }</pre>	
類別名稱	ECB
函式名稱	Connect
引數	Null
回傳	ECB_Errs 列舉 OK : 正常連接 Unconnected : TCP 通訊尚未建立 PackErr : ECB 封包錯誤 Others : 查看 log 檔判斷錯誤訊息
<p>此函式用於確認 ECB 是否正確連接，呼叫完 ECB_RunAsync 後使用。</p> <p>備註 : ECB_RunAsync & IsConnected 僅用於判斷 TCP 連線是否正常，Connect 函式用於判斷 ECB 是否被正確連接。</p>	

類別名稱	ECB
函式名稱	GetModule
引數	Null
回傳	Object[] Null：尚未連線或通訊失敗，查看 log 檔判斷錯誤訊息
<p>取得 ECB 的各模組資訊，例如 ECB 目前連接了兩張擷取卡，第一個插槽是 ADC 模組、第四個插槽是 TC 模組。當正確讀取後，moduleObj[0]永遠都是底板資訊，後續陣列內容則依照插卡順序，即 moduleObj[1]為 ADC 模組、moduleObj[2]為 TC 模組。</p> <p>注意：所有擷取模組的設定皆會透過此函式回傳的物件，詳細內容參考「擷取模組設定」。</p> <pre>//Connected...</pre> <pre>// ECB has ADC in slot 1 and TC in slot 4 object[] moduleObj = ecb.GetModule(); if(moduleObj != null) { //moduleObj[0] is bridge //moduleObj[1] is ADC module //moduleObj[2] is TC module }</pre>	
類別名稱	ECB
函式名稱	StartAcq
引數	unsigned int second：擷取秒數(預設為 0，代表無限制秒數)
回傳	ECB_Errs 列舉 OK：正常擷取 Unconnected：TCP 通訊尚未建立 NoECB：ECB 沒有正確連接 Others：查看 log 檔判斷錯誤訊息
<p>所有模組同步啟動擷取，如果沒有輸入引數，則會無限制時間進行擷取，引數單位為秒，當秒數到達便會自動停止。搭配 moduleObj[i].GetRecData()取得擷取的資料。</p>	

<pre>//Connected... // ECB has ADC in slot 1 object[] moduleObj = ecb.GetModule(); if(moduleObj != null) { ecb.StartAcq(10); thread.sleep(10000); //wait 10 second double[,] data = moduleObj[1].GetRecData(); }</pre>	
類別名稱	ECB
函式名稱	StopAcq
引數	Null
回傳	ECB_Errs 列舉 OK：正常停止 Unconnected：TCP 通訊尚未建立 NoECB：ECB 沒有正確連接 Others：查看 log 檔判斷錯誤訊息
<p>用於停止所有模組的擷取功能，當 StartAcq 沒有設定秒數或是想中途停止擷取皆可呼叫此函式。</p> <pre>//Connected... // ECB has ADC in slot 1 object[] moduleObj = ecb.GetModule(); if(moduleObj != null) { ecb.StartAcq(); thread.sleep(10000); //wait 10 second ecb.StopAcq(); double[,] data = moduleObj[1].GetRecData(); }</pre>	

類別名稱	ECB
函式名稱	ExportConfig
引數	Null
回傳	ECB_Errs 列舉 OK : 正常 NoECB : ECB 沒有正確連接 NoModules : ECB 沒有連接任何模組 Others : 查看 log 檔判斷錯誤訊息

此函式會將目前 ECB 的設定輸出成 Configuration.json 檔，如下圖，下次可使用 ImportConfig 設定 ECB。

```

    "EcbConfig": [
      {
        "Layer": 1,
        "ModuleType": 3,
        "Fs": 8789,
        "ChEnable": 6,
        "CalibratingVal": [
          2.315,
          85.23,
          -4.21,
          0.523
        ],
        "InputType": [
          0,
          1,
          1,
          0
        ],
        "SlotTitle": "ADC module",
        "ChTitle": [
          "Ch1",
          "Ch2",
          "Ch3",
          "Ch4"
        ]
      }
    ],
    {
      "Layer": 0,
      "ModuleType": 1,
      "Fs": 10,
      "ChEnable": 255,
      "CalibratingVal": [
        -2.521,
        0.52775,
        -8.22014,
        -0.523,
        0,
        0,
        0,
        0
      ],
      "InputType": [
        0,
        1,
        3,
        0,
        0,
        0,
        0,
        0
      ],
      "SlotTitle": "TC module",
      "ChTitle": [
        "Ch1",
        "Ch2",
        "Ch3",
        "Ch4",
        "Ch5",
        "Ch6",
        "Ch7",
        "Ch8"
      ]
    }
  ],
}
```

類別名稱	ECB
函式名稱	ImportConfig
引數	ref object[] moduleObj
回傳	ECB_Errs 列舉 OK：正常 NoECB：ECB 沒有正確連接 NoModules：ECB 沒有連接任何模組 NoConfigFile：沒有 Configuration.json 檔 Others：查看 log 檔判斷錯誤訊息
<p>用於導入先前產生的 Configuration.json 檔，減少 ECB 設定。注意需先連線成功後，呼叫 GetModule 後，將回傳值作為引數。</p> <pre>//Connected...</pre> <pre>object[] moduleObj = ecb.GetModule(); if(moduleObj != null) { ECB_Errs err = ImportConfig(ref moduleObj); }</pre>	
類別名稱	ECB
函式名稱	EnableInfluxDB
引數	Null
回傳	ECB_Errs 列舉 OK：正常 Others：查看 log 檔判斷錯誤訊息
<p>將 ECB 所記錄的資料皆上傳到 InfluxDB。</p> <p>注意：當此函式被呼叫後，便無法透過 moduleObj[i].GetRecData 取得資料，僅能透過 InfluxDB 網頁取得數據。</p> <p>注意：第一次啟用此功能前，需先使用 ExportConfig 輸出設定檔後，設定以下內容後方可使用此函式。</p> <pre>"InfluxDBConfig": { "IP": "127.0.0.1", "Port": 8086, "Token": "your token", "Organization": "your org", "Bucket": "your bucket", "Measurement": "your measurement" }</pre>	

● 擷取模組設定

以下函式或屬性皆屬於 `ecb.GetModule` 所回傳的物件。

類別名稱	EcbModule_Bridge, EcbModule_TC, EcbModule_RTD, EcbModule_ADC
屬性名稱	ModuleType
值	EcbModule.Module_Type 列舉 No_conn = 0 : 未連接 TC = 1 : 熱電偶模組 RTD = 2 : PT100/PT1000 模組 ADC = 3 : ADC 模組 Bridge = 0xFF : 底板

此屬性可用於判斷擷取卡類型，例如 ECB 目前連接了兩張擷取卡，第一個插槽是 ADC 模組、第四個插槽是 TC 模組。

```
//Connected...
```

```
// ECB has ADC in slot 1 and TC in slot 4
```

```
object[] moduleObj = ecb.GetModule();
```

```
if(moduleObj != null)
```

```
{
```

```
    var type = (EcbModule)moduleObj[0].ModuleType; // type ==  
    Bridge(0xFF)
```

```
    type = (EcbModule)moduleObj[1].ModuleType; // type == ADC(3)
```

```
    type = (EcbModule)moduleObj[2].ModuleType; // type == TC(1)
```

```
}
```

類別名稱	EcbModule_TC, EcbModule_RTD, EcbModule_ADC
函式名稱	SetFs
引數	unsigned short
回傳	ECB_Errs 列舉 OK：設定成功 PackErr：封包傳送錯誤 Timeout：傳輸逾時
<p>此函式用於設定擷取模組的擷取頻率，單位為 Hz。</p> <p>注意：各模組最大值如下</p> <p>TC：10Hz</p> <p>RTD：60Hz</p> <p>ADC：52,734Hz</p> <p>Ex：ADC 模組於插槽 1</p> <pre>//Connected...</pre> <pre>// ECB has ADC in slot 1 object[] moduleObj = ecb.GetModule(); if(moduleObj != null) { // 設定取樣頻率為 1kHz ECB_Errs err = moduleObj[1].SetFs(1000); if(err == ECB_Errs.OK) { // 取得實際運作的取樣頻率 unsigned short realFs = moduleObj[1].Fs; } }</pre>	
類別名稱	EcbModule_TC, EcbModule_RTD, EcbModule_ADC
函式名稱	SetPreTrigger
引數	Unsigned int (預設為 0)
回傳	ECB_Errs 列舉 OK：設定成功 PackErr：封包傳送錯誤 Timeout：傳輸逾時
<p>此函式用於設定擷取模組的預觸發資料筆數。</p> <p>注意：各模組最大值如下</p> <p>TC & RTD：512,000 筆</p> <p>ADC：85,333 筆</p>	

類別名稱	EcbModule_TC, EcbModule_RTD, EcbModule_ADC
函式名稱	RemoveInput
引數	Null
回傳	ECB_Errs 列舉 OK：設定成功 PackErr：封包傳送錯誤 Timeout：傳輸逾時
<p>移除所有頻道輸入，建議在每次設定 ECB 前，都先呼叫此函式，避免有先前設定的輸入頻道。</p> <pre>//Connected... // ECB has ADC in slot 1 object[] moduleObj = ecb.GetModule(); if(moduleObj != null) { // 移除所有輸入頻道 ECB_Errs err = moduleObj[1].RemoveInput(); if(err == ECB_Errs.OK) { // 移除成功，可以開始設定需要的頻道 } }</pre>	
類別名稱	EcbModule_TC, EcbModule_RTD, EcbModule_ADC
函式名稱	RemoveInput
引數	Byte：想要移除的頻道編號(start from 0)
回傳	ECB_Errs 列舉 OK：設定成功 PackErr：封包傳送錯誤 Timeout：傳輸逾時
<p>移除指定的頻道輸入。</p> <p>例如：想要移除 Ch1 以及 Ch3 輸入</p>	


```
//Connected...

// ECB has ADC in slot 1
object[] moduleObj = ecb.GetModule();
if(moduleObj != null)
{
    // 移除 Ch1
    ECB_Errs err = moduleObj[1].RemoveInput(0);
    // 移除 Ch3
    err = moduleObj[1].RemoveInput(2);
}
```

類別名稱	EcbModule_TC, EcbModule_RTD, EcbModule_ADC
函式名稱	AddInput
引數	Byte：想要新增的輸入頻道(start from 0) InputType 列舉：輸入類型
回傳	ECB_Errs 列舉 OK：設定成功 PackErr：封包傳送錯誤 Timeout：傳輸逾時

新增輸入頻道，並且設定輸入感測器類型。

例如：想新增熱電偶模組的 Ch2 以及 Ch5，且使用 type E 熱電偶

```
//Connected...

// ECB has TC in slot 1
object[] moduleObj = ecb.GetModule();
if(moduleObj != null)
{
    // 移除所有輸入頻道
    ECB_Errs err = moduleObj[1].RemoveInput();
    if(err == ECB_Errs.OK)
    {
        // 新增 Ch2，感測器為 E type
        err = moduleObj[1].AddInput(1, EcbModule_TC.InputType.E);
        // 新增 Ch5，感測器為 E type
        err = moduleObj[1].AddInput(4, EcbModule_TC.InputType.E);
    }
}
```

```
/// <summary>
/// Input type of TCmodule.
/// </summary>
public enum InputTypes : byte
{
    B = 0, E = 1, J = 2, K = 3, N = 4, R = 5, S = 6, T = 7
};

/// <summary>
/// Input type of RTD module.
/// </summary>
public enum InputTypes : byte
{
    PT100 = 0, PT1000 = 1
};

/// <summary>
/// Input type of ADC module.
/// </summary>
public enum InputTypes : byte
{
    ADC = 0, IEPE = 1
};
```

類別名稱	EcbModule_TC, EcbModule_RTD, EcbModule_ADC
函式名稱	StartAcq
引數	Unsigned int：擷取秒數
回傳	ECB_Errs 列舉 OK：設定成功 PackErr：封包傳送錯誤 Timeout：傳輸逾時 Others：查看 log 檔判斷錯誤訊息
<p>開始擷取資料。</p> <p>注意：ECB 物件的 StartAcq 方法是同步啟動所有的擷取卡，此處的 StartAcq 僅針對單一擷取卡。</p> <pre>//Connected... // ECB has TC in slot 1 object[] moduleObj = ecb.GetModule(); if(moduleObj != null) { // 移除所有輸入頻道 ECB_Errs err = moduleObj[1].RemoveInput(); if(err == ECB_Errs.OK) { // 新增 Ch2，感測器為 E type moduleObj[1].AddInput(1, EcbModule_TC.InputType.E); // 新增 Ch5，感測器為 E type moduleObj[1].AddInput(4, EcbModule_TC.InputType.E); // 啟動擷取 err = moduleObj[1].StartAcq(); // 等待 10 秒 thread.sleep(10000); // 停止擷取 moduleObj[1].StopAcq(); // 取得資料 var data = moduleObj[1].GetRecData(); } }</pre>	

類別名稱	EcbModule_TC, EcbModule_RTD, EcbModule_ADC
函式名稱	StopAcq
引數	Null
回傳	ECB_Errs 列舉 OK：設定成功 PackErr：封包傳送錯誤 Timeout：傳輸逾時 Others：查看 log 檔判斷錯誤訊息
<p>停止擷取資料。</p> <p>注意：ECB 物件的 StartAcq 方法是同步啟動所有的擷取卡，此處的 StartAcq 僅針對單一擷取卡。</p>	
類別名稱	EcbModule_TC, EcbModule_RTD, EcbModule_ADC
函式名稱	GetRecData
引數	Null
回傳	Double[,]：目前已記錄的資料
<p>取得目前已擷取的資料。</p> <p>TC、RTD 資料單位：0.001°C</p> <p>ADC 資料單位：10 / (2²⁴ - 1) Volt</p>	
類別名稱	EcbModule_TC, EcbModule_RTD, EcbModule_ADC
函式名稱	SetCalibratingVal
引數	Float[]：頻道校正值
回傳	ECB_Errs 列舉 OK：設定成功 PackErr：封包傳送錯誤 Timeout：傳輸逾時 Others：查看 log 檔判斷錯誤訊息
<p>設定各頻道校正值，陣列的每一元素皆代表每一個頻道的校正值。</p> <p>例如：TC 模組的 Ch1 校正值為 0.32，Ch3 校正值為-1.21，其餘為 0</p>	

```
//Connected...

// ECB has TC in slot 1
object[] moduleObj = ecb.GetModule();
if(moduleObj != null)
{
    float[] data = new float[8]{0};
    data[0] = 0.32;
    data[2] = -1.21;
    // 設定校正値
    ECB_Errs err = moduleObj[1].SetCalibratingVal(data);
}
```