

# ESD-01

## 嵌入式系統設計

陳慶瀚

MIAT嵌入式計算實驗室/中央大學資工系

[pierre@csie.ncu.edu.tw](mailto:pierre@csie.ncu.edu.tw)

2009年9月18日



# 本課程內容

- 一 嵌入式系統概論(架構、方法、技術)
- 一 嵌入式系統設計方法論
- 一 嵌入式硬體實作
- 一 嵌入式硬體期末專題





# 嵌入式系統是什麼？

- 定義：作為機器、儀控設備、消費型電子產品一部分的 **computing system**，透過感測(sensor)與制動(actuator)與環境互動，稱為嵌入式系統。
- 一個嵌入式系統具有以下特色：
  1. *Single-functioned*
  2. *Tightly constrained(size,power,cost,perform.)*
  3. *Reactive and real-time*



# 電腦V.S.嵌入式系統

特徵	泛用型電腦	嵌入式系統
類型	<ul style="list-style-type: none"><li>• 看得見的電腦。</li><li>• 按其結構、运算速度和結構規模分類。</li></ul>	<ul style="list-style-type: none"><li>• 看不見的電腦。</li><li>• 形式多樣，應用領域廣泛，無法分類，通常以應用來區分</li></ul>
組成	<ul style="list-style-type: none"><li>• 泛用型處理器、標準匯流排和外部週邊。</li><li>• 軟體和硬體相對獨立。</li></ul>	<ul style="list-style-type: none"><li>• 應用導向的嵌入式微處理器，匯流排和週邊緣件多整合在處理器內部。</li><li>• 軟體與硬體緊密結合</li></ul>
開發方式	<ul style="list-style-type: none"><li>• 開發平臺和應用執行平臺都是泛用型電腦</li></ul>	<ul style="list-style-type: none"><li>• 採用cross development，開發平臺是泛用型電腦，運行應用平臺是target嵌入式系統。</li></ul>
二次開發	<ul style="list-style-type: none"><li>• 應用程式可reprogrammable</li></ul>	<ul style="list-style-type: none"><li>• 一般無法reprogrammable</li></ul>



# 嵌入式硬體

## Processors

- Application-Specific Integrated Circuit (ASIC)
- Application-Specific Standard Product (ASSP)
- Field-Programmable Gate Array (FPGA)
- Microcontroller (MCU)
- Digital Signal Processor (DSP)
- Hybrid DSP/MCU

## Peripherals

- Real Time Timer/Watchdog • LCD • Keypad
- Motor • UART • ADC/DAC • PWM
- Sensor...



# 嵌入式軟體

## Device Driver

USB, Image sensor, LCD, Keyboard, Bluetooth, Wifi, Touch panel, GPS, ...

## Real-Time OS

QNX, RTLinux, VxWorks, Windows CE, OS-9 ...  
MicroC/OS-II, eCos, RTLinux, freeRTOS ...

## Middleware

Embedded GUI, 2D/3D Graphics, Imaging, Wireless ...

## Application

Robot, Portable Game, Media Player, Biomedical Intrum., DSC...



# 嵌入式系統的產品趨勢

- 技術推陳出新
- 多樣化生產形式
- 產品生命週期短
- 單價低
- 內建一個以上的微處理器

>>Evolving to Complexity!



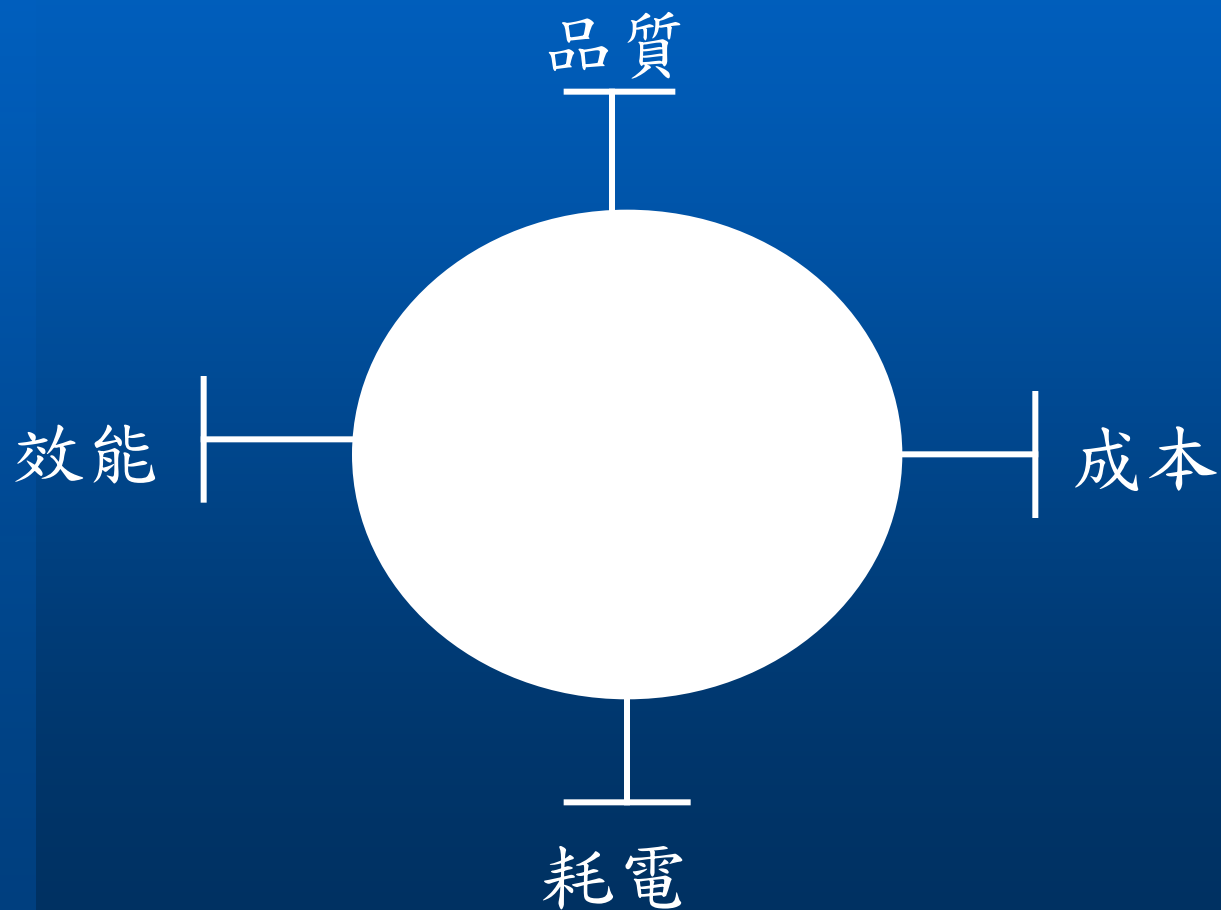


# 嵌入式系統設計挑戰

- 功能正確性
- 單位製造成本
- NRE(Non-Recurring Engineering)費用
- 體積
- 效能
- 耗電
- 彈性
- 原型產品產出時間(Time-to-prototype)
- 上市時間(Time-to-market)
- 維修性
- 安全性

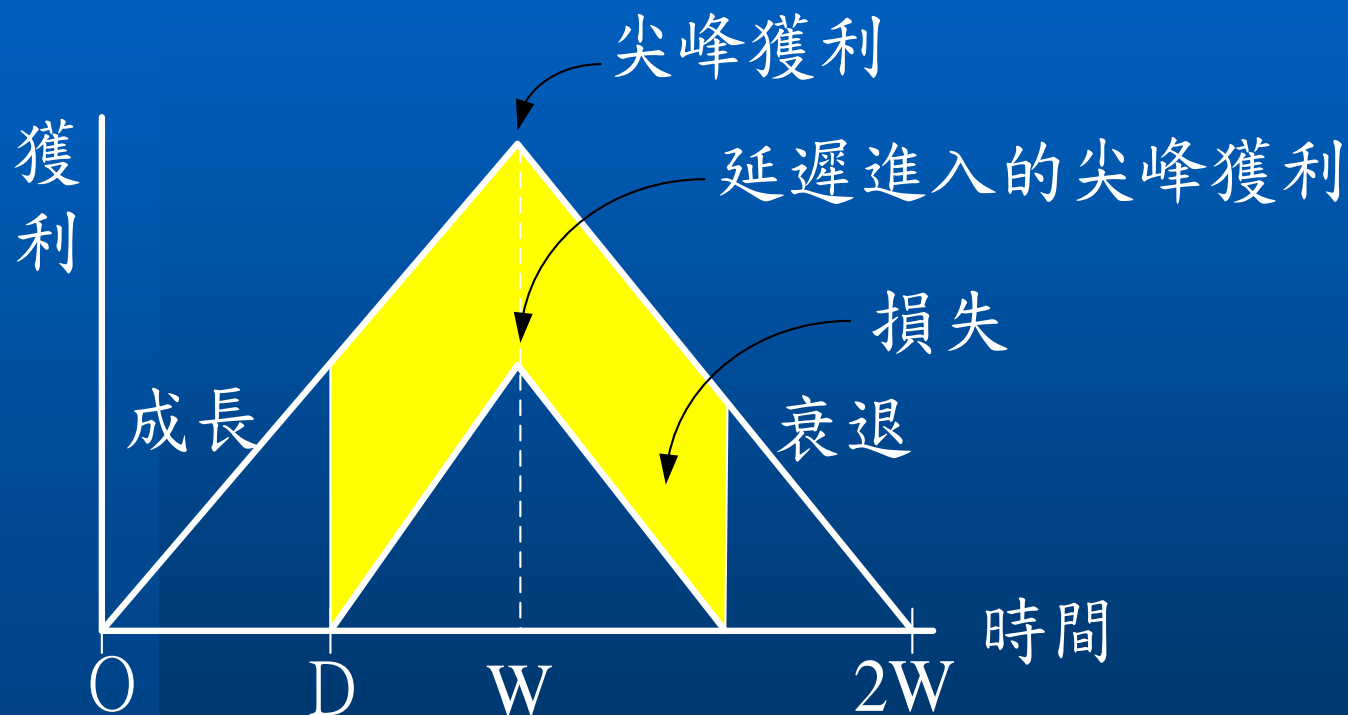


# 嵌入式系統設計挑戰





# Time-to-market



- 產品生命週期  $2W=52$  周, 延遲進入 4 周  $D=4$ , 損失 = 22%
- 產品生命週期  $2W=52$  周, 延遲進入 10 周  $D=4$ , 損失 = 50%



# 嵌入式系統設計主要技術

- 微處理器(Microprocessor): 泛用型處理器, 專用型處理器
- 晶片(IC): 全客戶型IC, 半客戶型IC, 可程式IC
- 設計方法論(Design Methodology): 設計、編譯、電路合成、測試、驗證的工具、技術和方法



# 嵌入式系統產品設計

- 軟/硬體平台密切配合
  - 需要充分了解硬體特點，特別是處理器的性能。
- 選擇正確的開發工具
  - 快速開發軟體，且程式碼移植性高。
- 程式執行效率高，程式碼精簡。
- 消費性電子產品需提供方便的使用者界面。

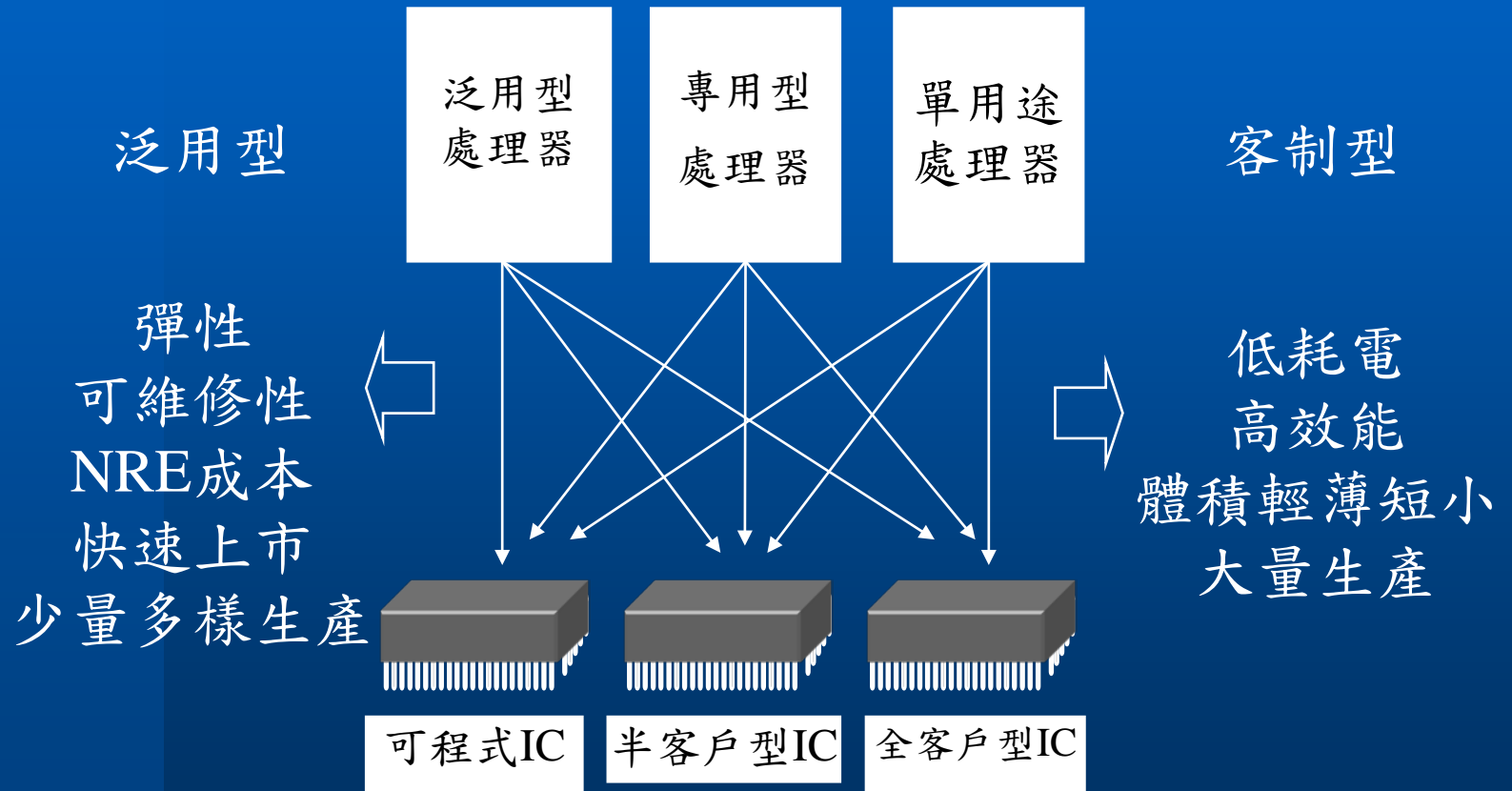


# 嵌入式系統產品設計流程

- Product Requirement Definitions
- Functionality Description
- Processor Selection
- Programming/PCB Layout
- Hardware Design/Verification
- Firmware Design/Testing
- Integration

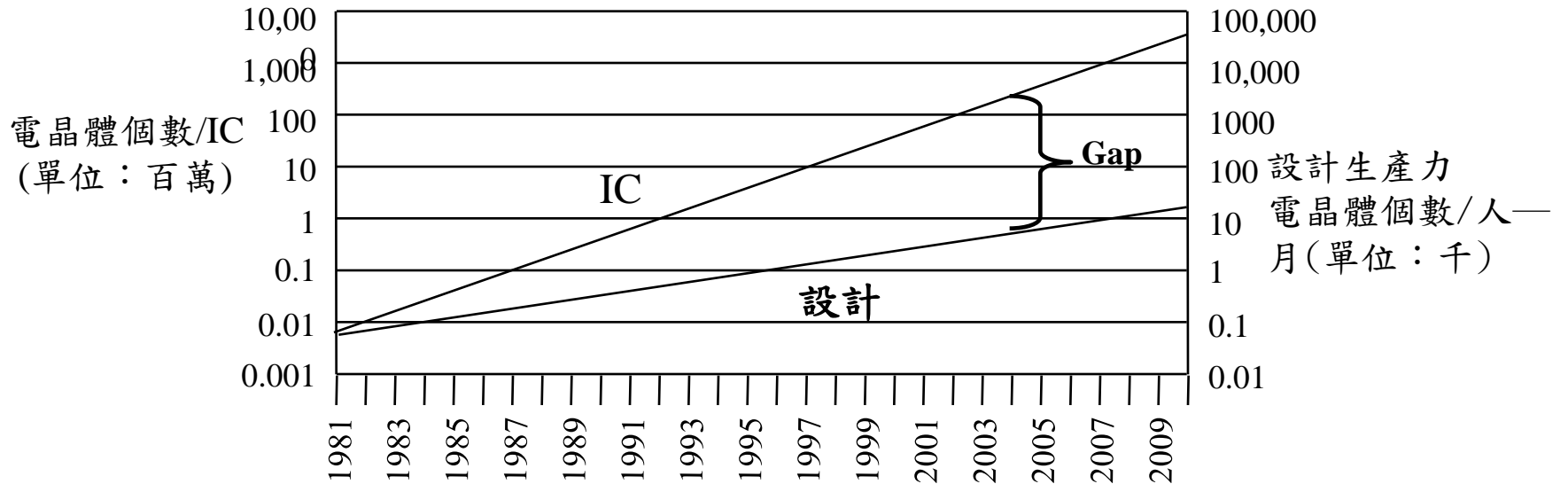


# 設計選擇





# 為何需要設計方法論？







# 嵌入式系統成本

- **Unit cost**: the monetary cost of manufacturing each copy of the system, excluding NRE cost
- **NRE cost** (Non-Recurring Engineering cost): The one-time monetary cost of designing the system
- $total\ cost = NRE\ cost + unit\ cost * \#\ of\ units$
- *per-product cost*
  - =  $total\ cost / \#\ of\ units$
  - =  $(NRE\ cost / \#\ of\ units) + unit\ cost$



# Why 嵌入式系統設計方法論？

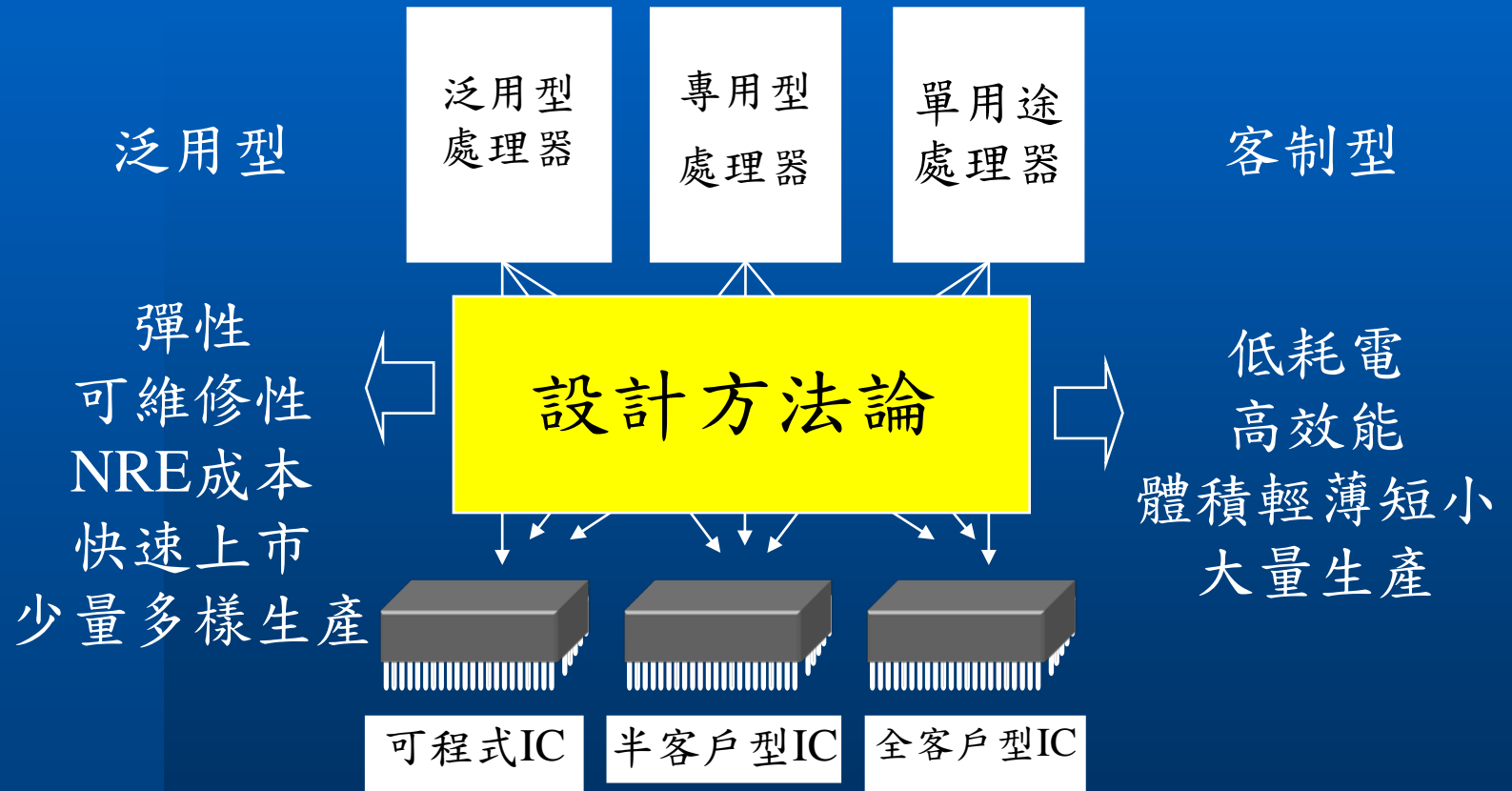
- 建構一個知識管理平台，使個體的經驗和知識得以累積和分享；
- 保護組織的研發知識成果；
- 運用群體知識，避免個人知識解題的侷限性；
- 避免重蹈錯誤和資源重複投資。



# 嵌入式系統設計方法論

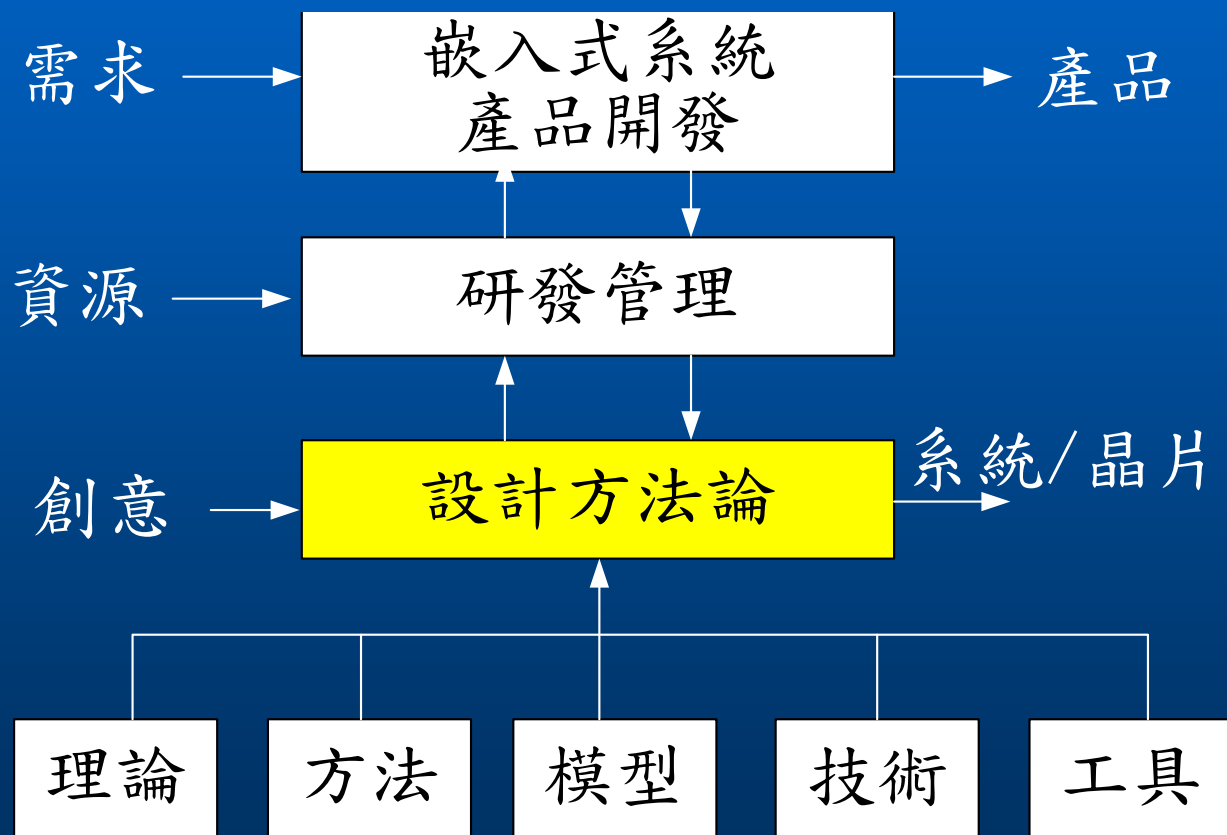


# 設計Trade-Off



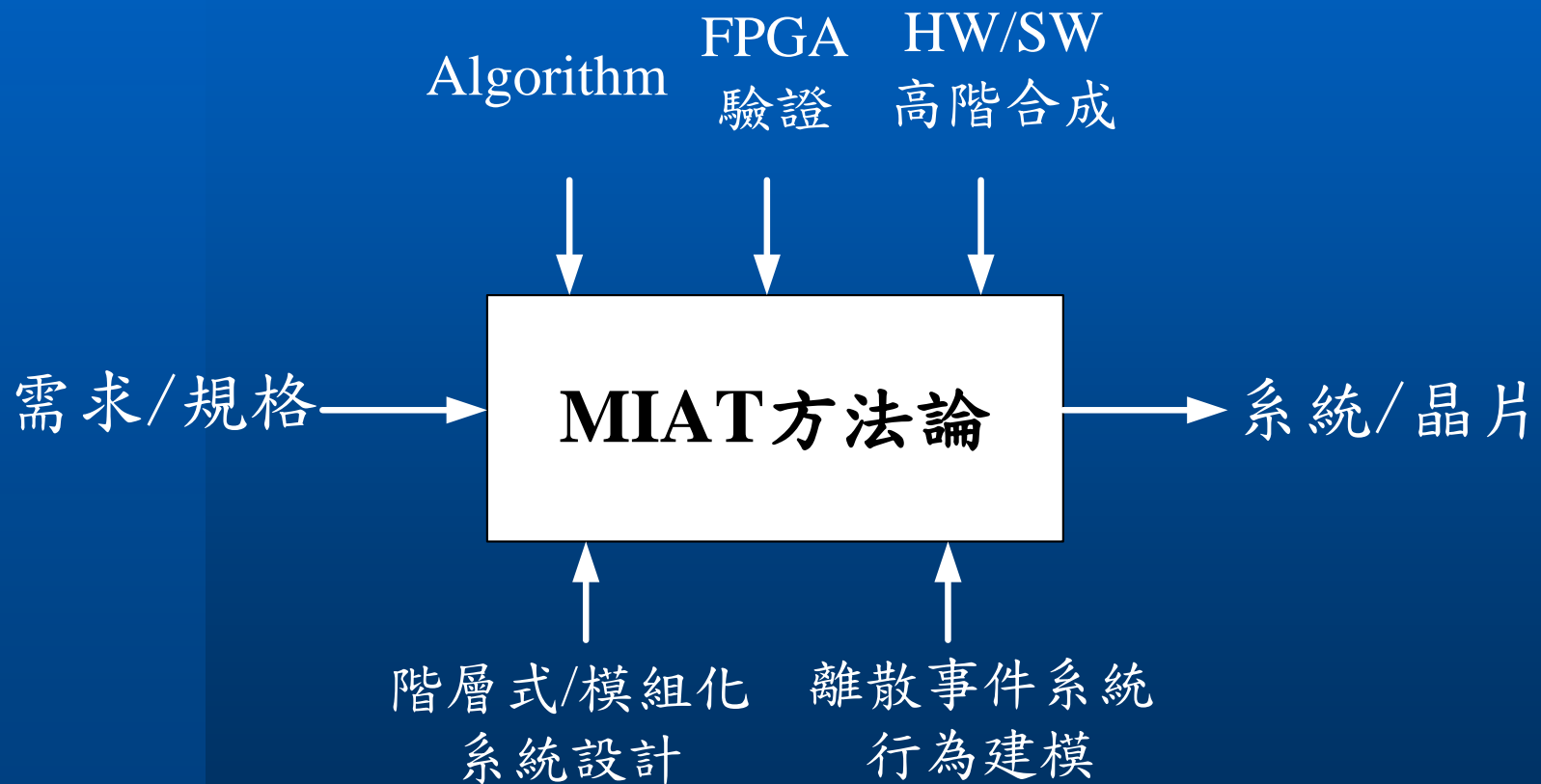


# 開發鏈的新角色—設計方法論





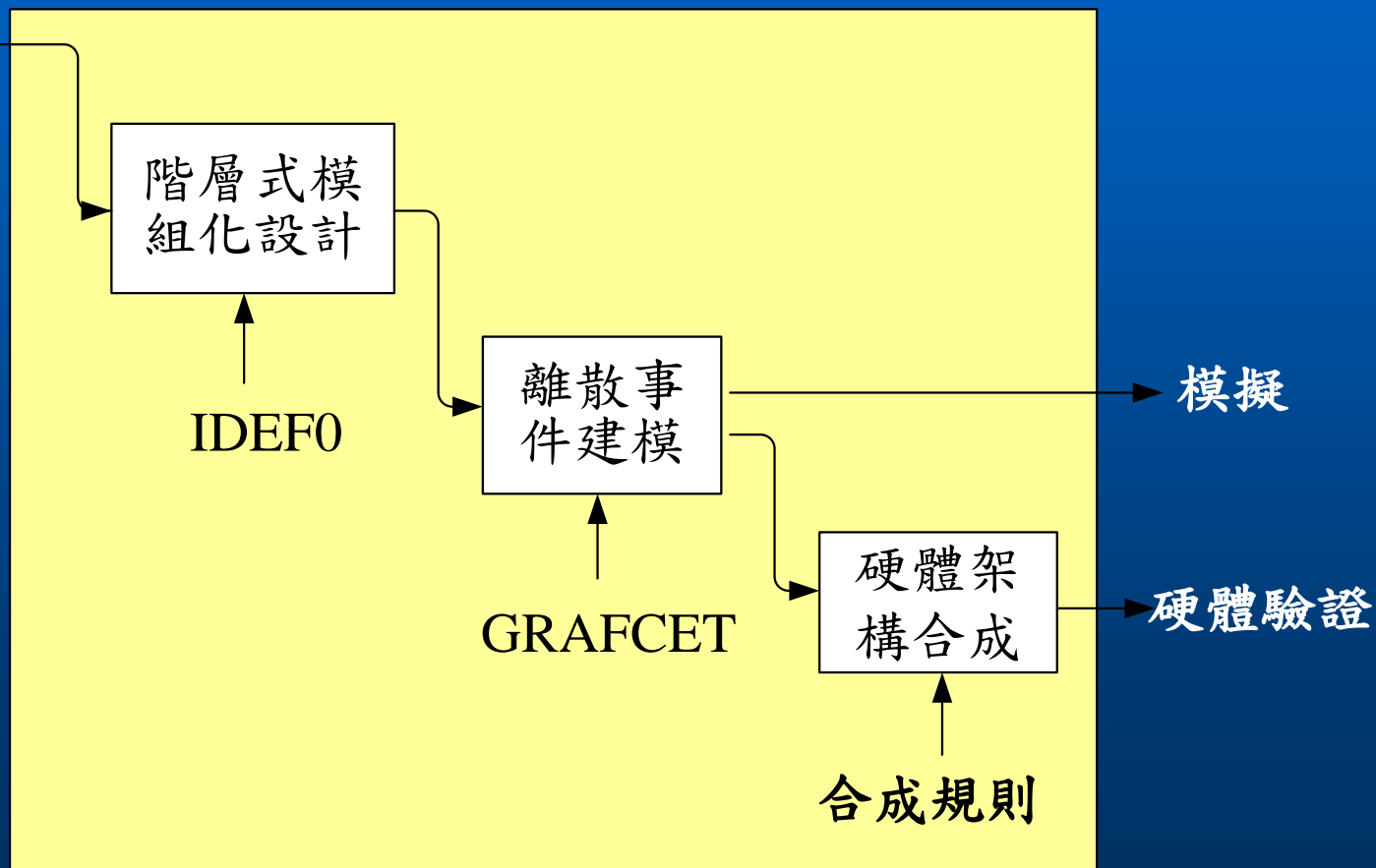
# MIAT複雜系統設計方法論





# MIAT方法論上層架構

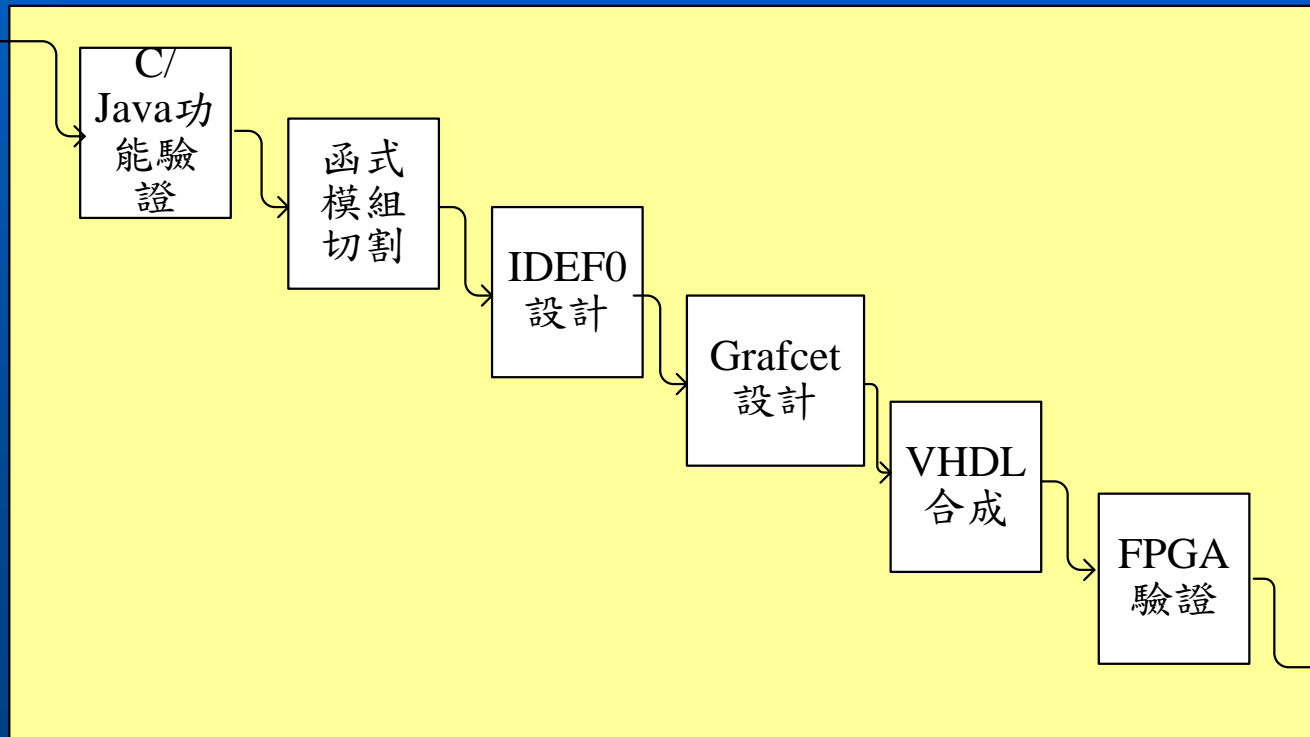
規格定義





# MIAT方法論的底層架構

方法、  
演算法

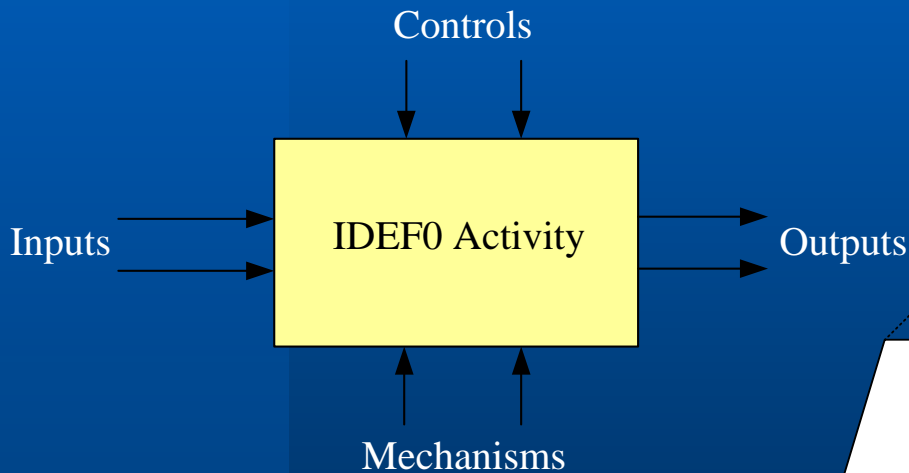




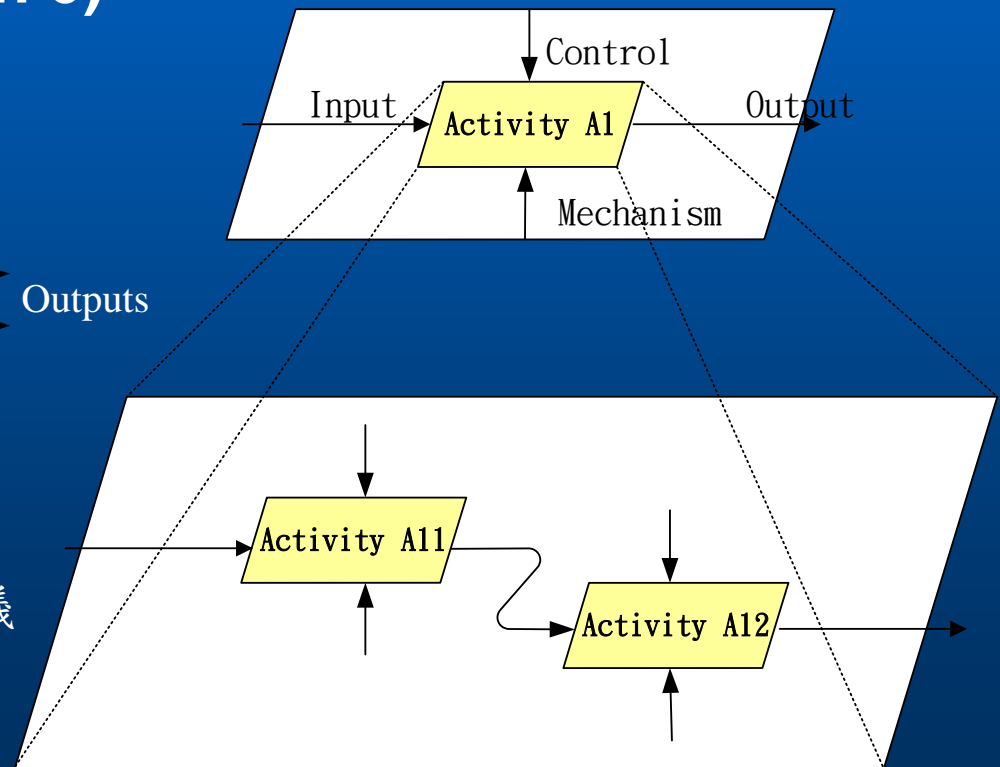


# 階層式與模組化設計

- IDEF0(Integrated Computer-Aided Manufacturing (ICAM) DEFinition 0)



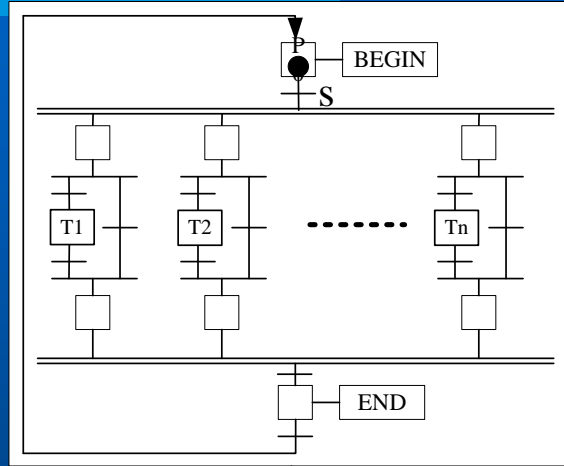
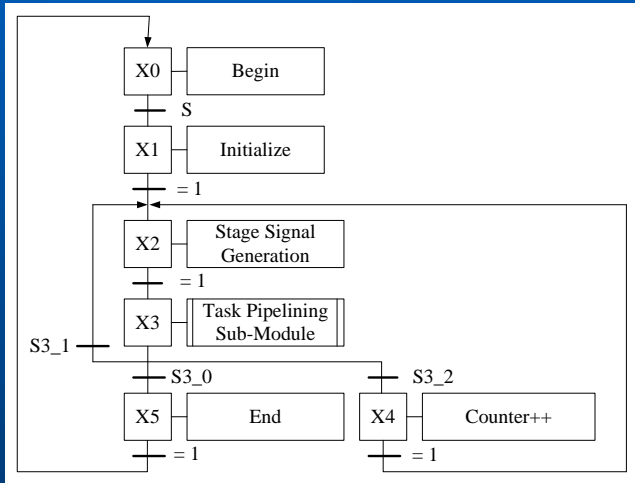
IDEF0基本模組功能方塊與箭頭意義



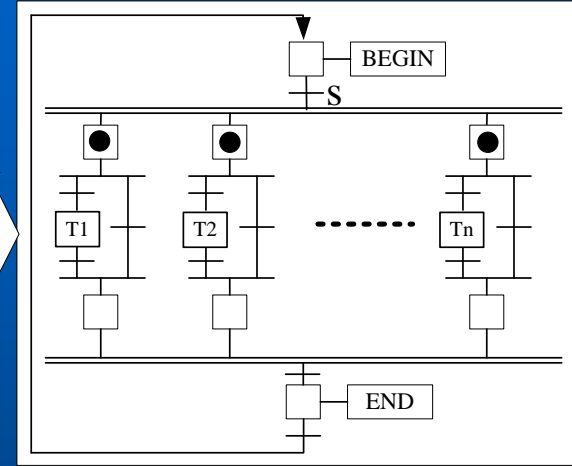
IDEF0 階層化架構



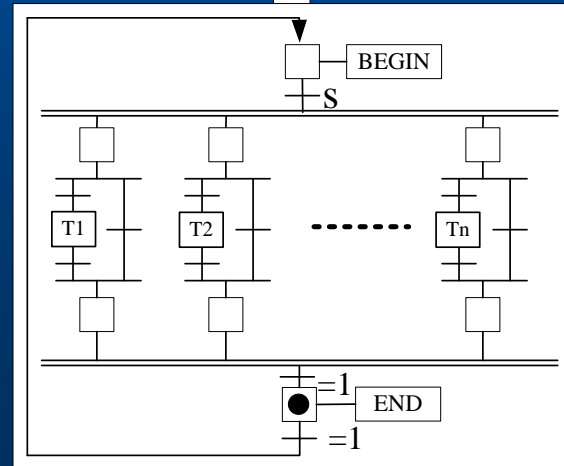
# GRAFCET 離散事件建模



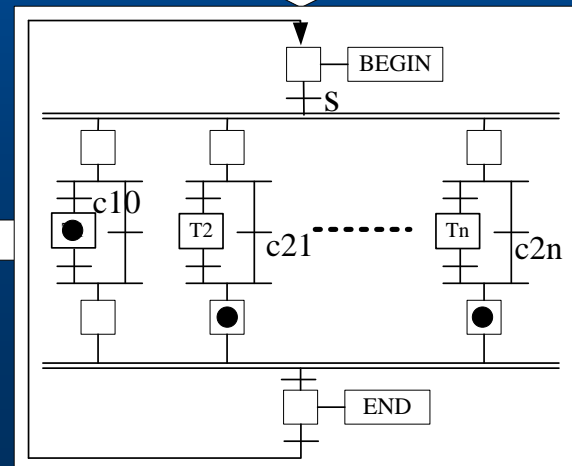
S=1



c10=1, c12=1,...



S=1







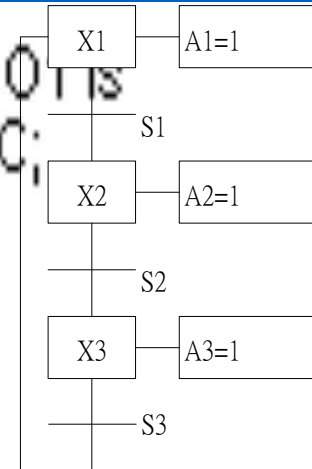
# VHDL硬體合成

```
architecture DEMO1_arch of DEMO1 is  
    signal X1,X2,X3:STD_LOGIC;  
begin  
    process(CLK)  
    begin  
        if RESET='1' then
```

```
            X1<='1';  
            X2<='0';  
            X3<='0';
```

```
        elsif CLK'EVENT and CLK='1' then
```

```
            if X1='1' and S1='1' then X
```







# 方法論工作流程

## 1. C/C++演算法驗證

C/C++程式碼, 驗證資料檔, 離散版本演算法

## 2. IDEFO階層式模組架構圖設計

## 3. GRAFCET離散事件模型建立

## 4. VHDL電路合成

## 5. 硬體驗證

時序模擬波形, 電路方塊圖, 效能/使用資源檔

## 6. 命名規範

## 7. 文件撰寫



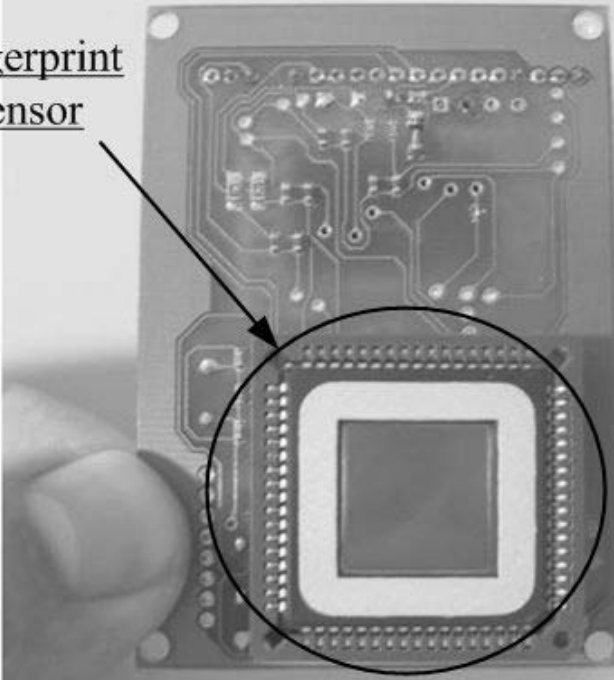
# MIAT嵌入式系統晶片設計實例

- 精簡嵌入式指紋辨識系統
- 視覺伺服自動倒車入庫系統
- 智慧型影像插補器ASIC
- 渾沌加密與保密通訊ASIC
- 被動式自動對焦系統
- 具保密通信功能的指紋遙控器
- DSP嵌入式指紋辨識系統
- 仿昆蟲智慧型六腳機器人

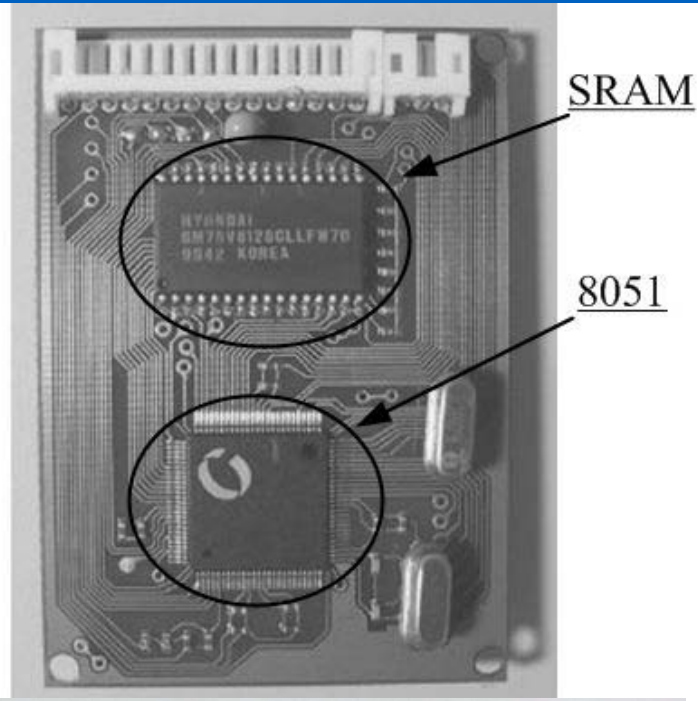


# 精簡嵌入式指紋身份識別系統

Fingerprint  
Sensor



SRAM

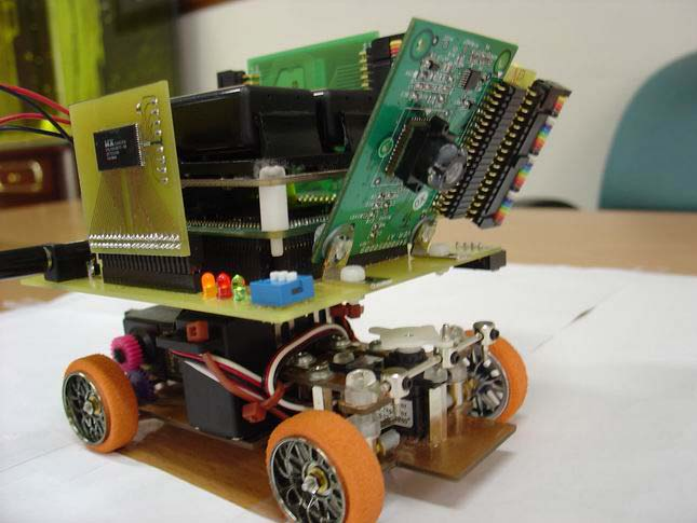


8051

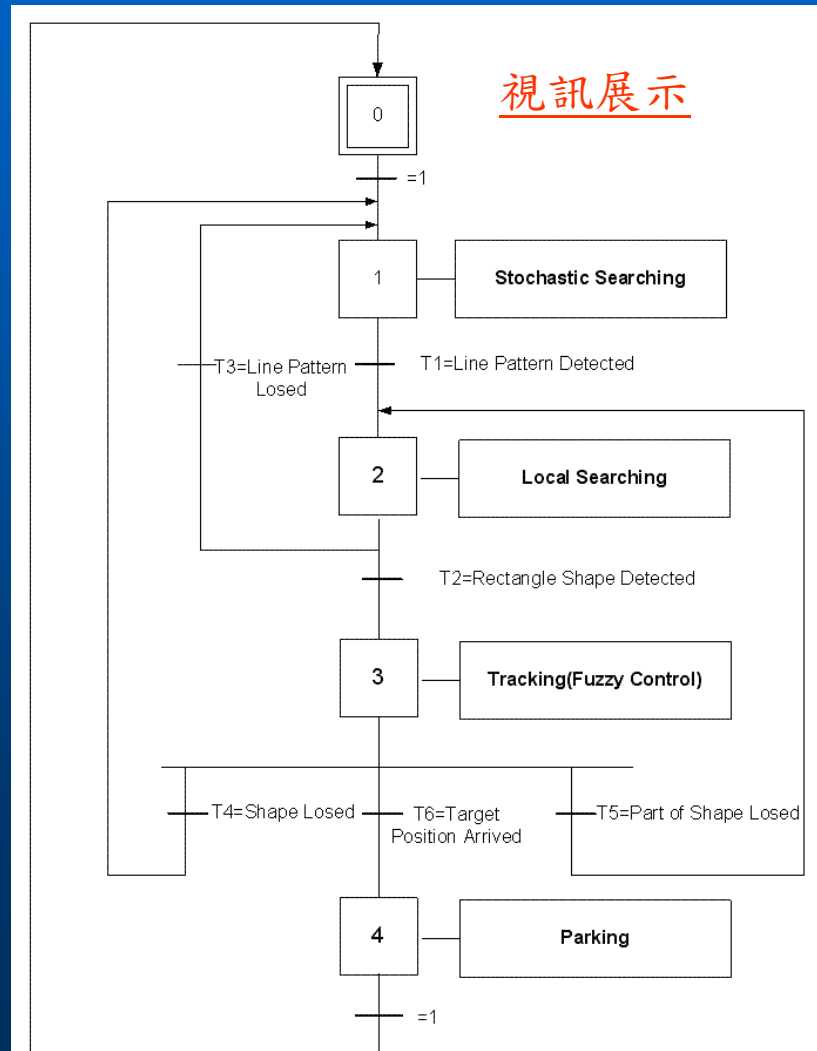
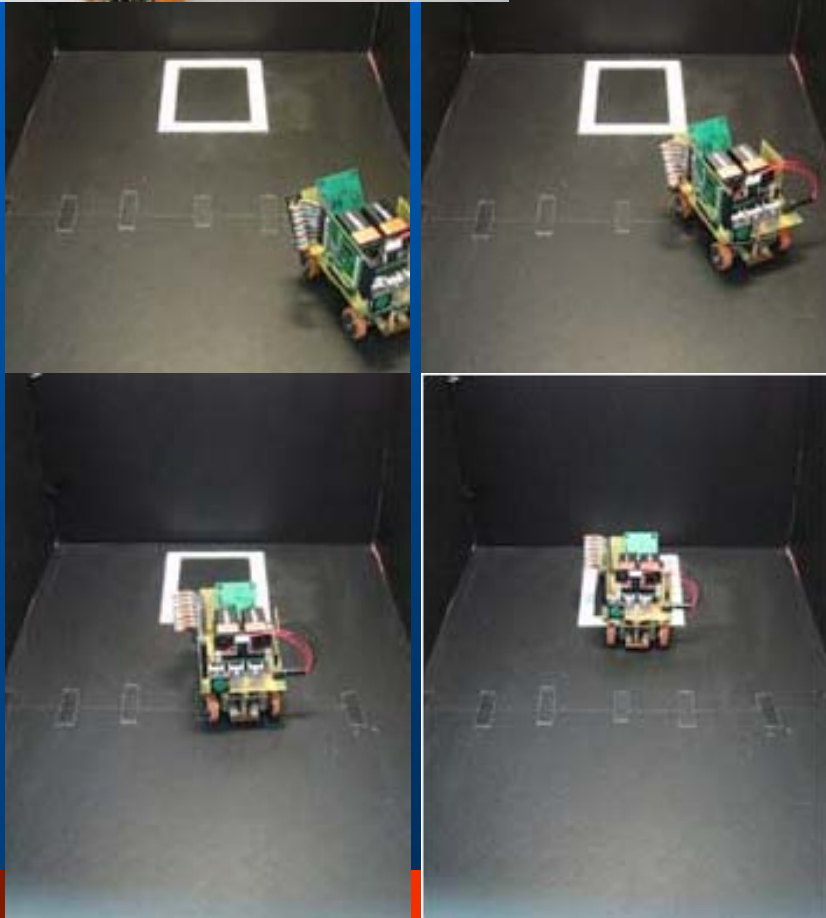
**25MHz 8051**  
**12 Kbytes ROM**  
**35 Kbytes RAM**  
**EER=4.5%**







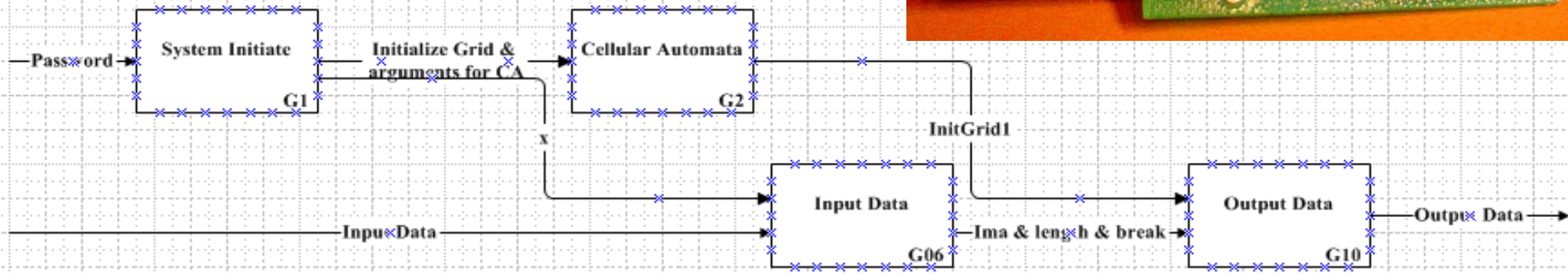
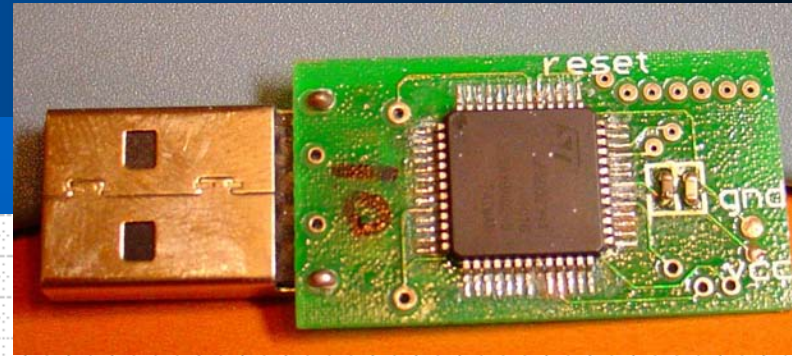
# 視覺伺服自動倒車入庫系統



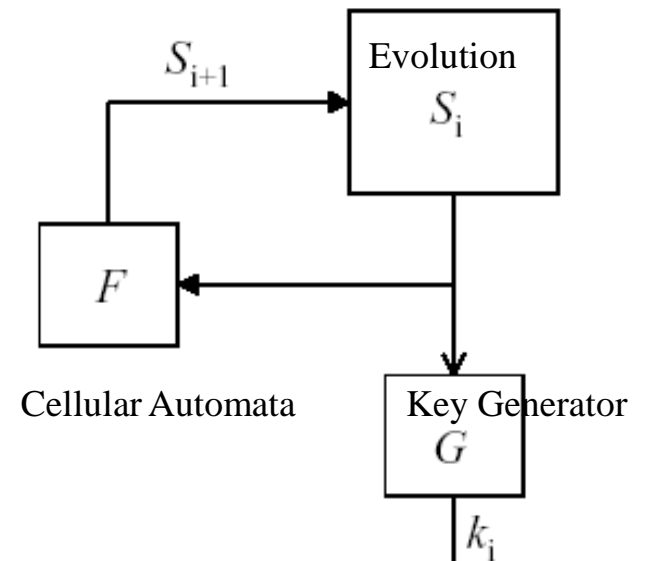




# 渾沌加密晶片

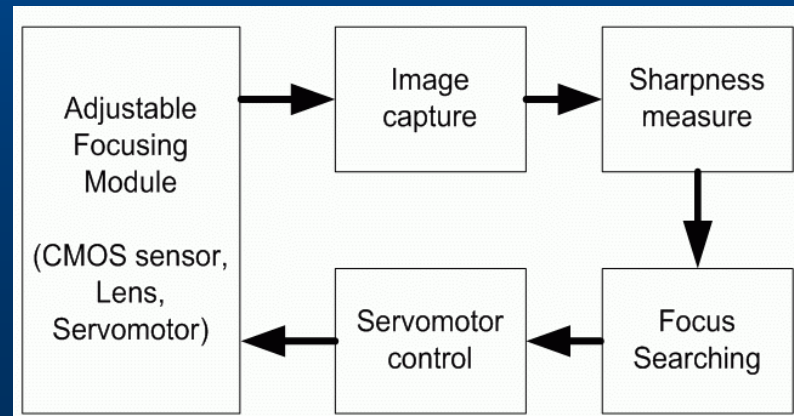
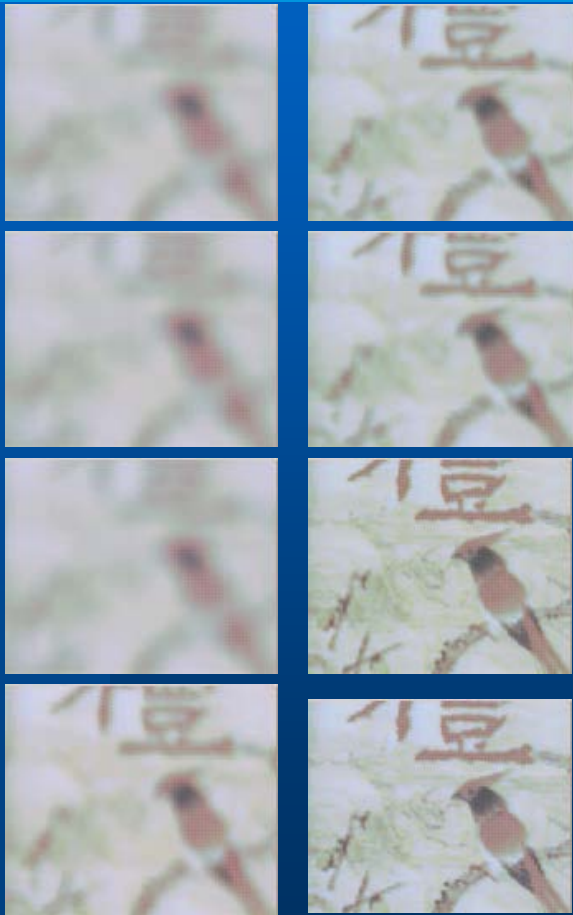


Gate Counts: ~ 70,000  
Speed: 41.6 MB/s at 130 MHz  
應用：加密影音撥放器/加密隨身碟/無線網路保密通訊/FPGA 電路保密裝置





# 被動式自動對焦系統



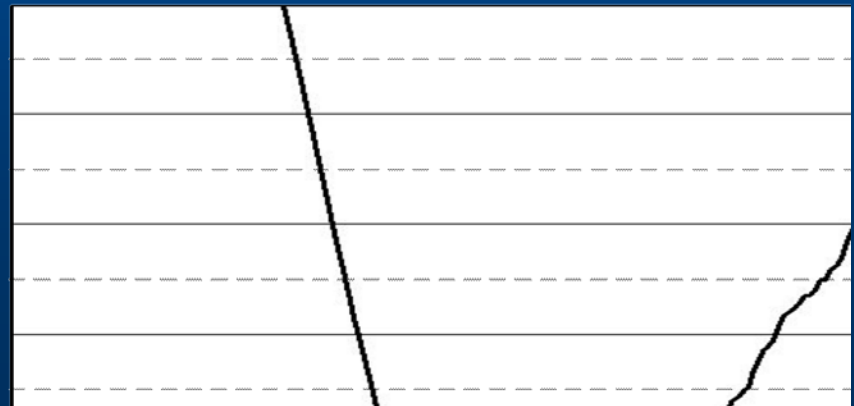


# DSP嵌入式指紋辨識系統



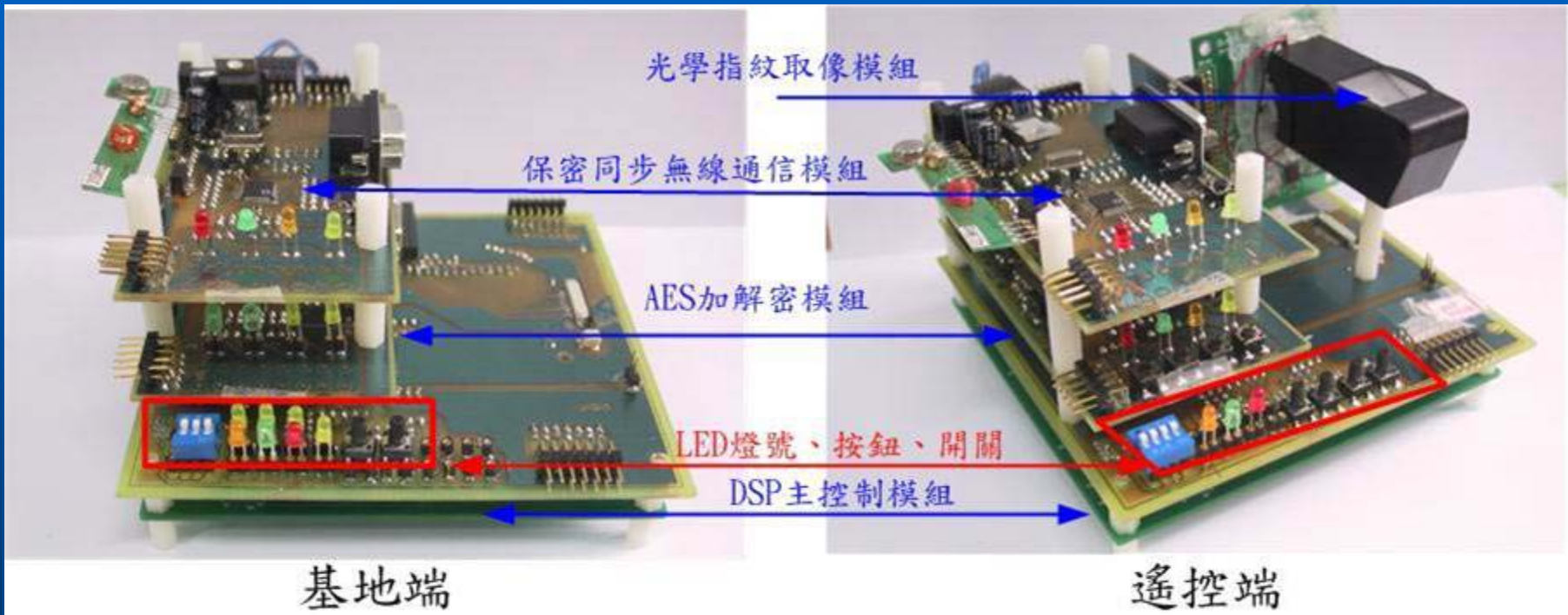
54x13=702枚指紋EER=1.41%  
辨識速度<0.5秒/一枚指紋。  
Code Memory:33KBytes  
Data Memory:169KBytes  
Flash Memory: 480 bytes/指紋特徵  
指紋特徵比對速度：800枚/sec

35%  
%  
25  
10%  
20  
%  
5%  
15  
%

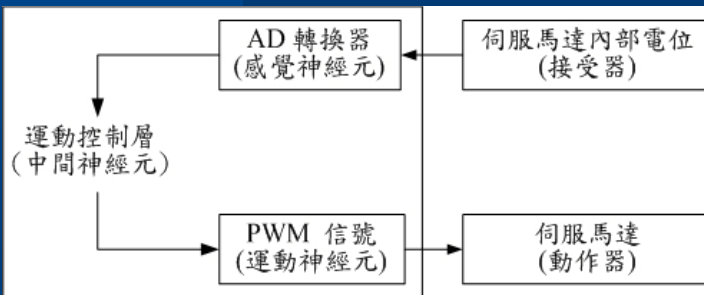
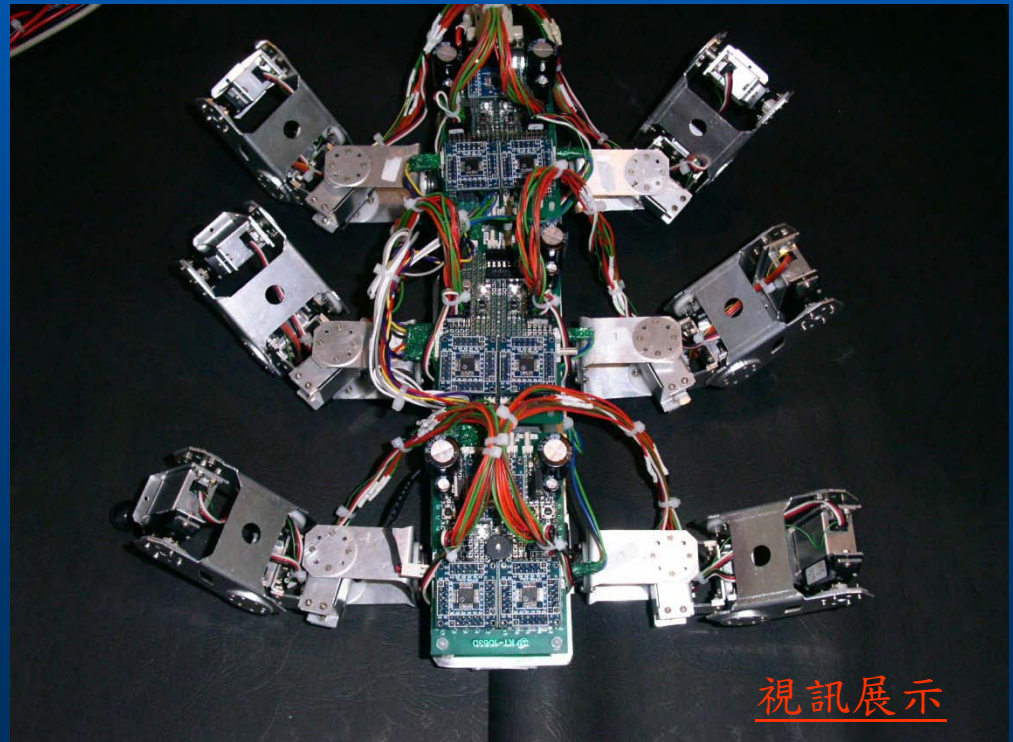
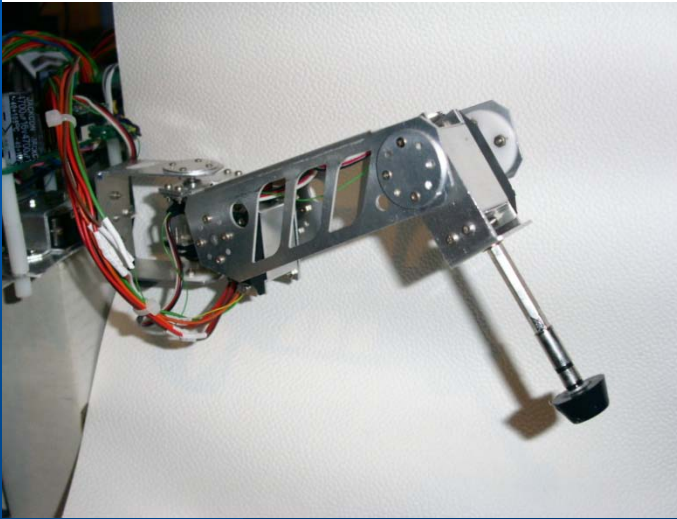
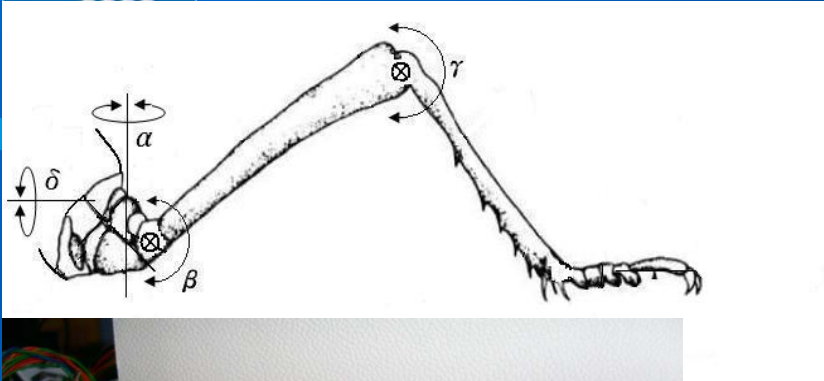




# 具保密通信功能的指紋遙控器



# 仿昆蟲智慧型六腳機器人



視訊展示



# MIAT嵌入式系統設計方法論

- MIAT設計方法論是基於Top-Down設計範式(paradigm)，系統設計採階層式、模組化的功能架構；
- 針對每一個功能獨立的模組，使用GRAFCET圖形化工具建立其離散事件模型；
- 根據一組合成的法則，所有GRAFCET模型得以轉譯為VHDL控制器電路，再結合一些通用的資料流(data flow)元件，便可實現系統的高階合成；
- 方法論採用均一的解決問題程序，包括問題形式化(formalization)、建模(modeling)和設計規則、工具和技術。因此利於知識的整合和團隊研發。





# 結語