

降壓式轉換器實驗模組

PEK-120

使用手冊



ISO-9001 認證企業

GW INSTEK

本手冊所含資料受到版權保護，未經固緯電子實業股份有限公司預先授權，不得將手冊內任何章節影印、複製或翻譯成其它語言。

本手冊所含資料在印製之前已經過校正，但因固緯電子實業股份有限公司不斷改善產品，所以保留未來修改產品規格、特性以及保養維修程式的權利，不必事前通知。

目錄

簡介	4
Buck DC-DC Converter 模組教具簡介	6
教學實驗平台	10
實驗目標	11
本書章節說明	12
PSIM 簡介	14
教具硬體及設備說明	17
功率級電路	17
DSP 控制電路	23
輔助電源	26
驅動電路	27
JTAG 燒錄電路	29
實驗設備說明	30
實驗 1 脈波寬度調變 (PWM)	36
實驗目的	36
實驗原理	37
電路模擬	54
SimCoder 程式規劃	55
實驗量測	69
實驗 2 電壓模式控制之降壓式轉換器	72
實驗目的	72
實驗原理	72
電路模擬	86
SimCoder 程式規劃	90
實驗量測	96

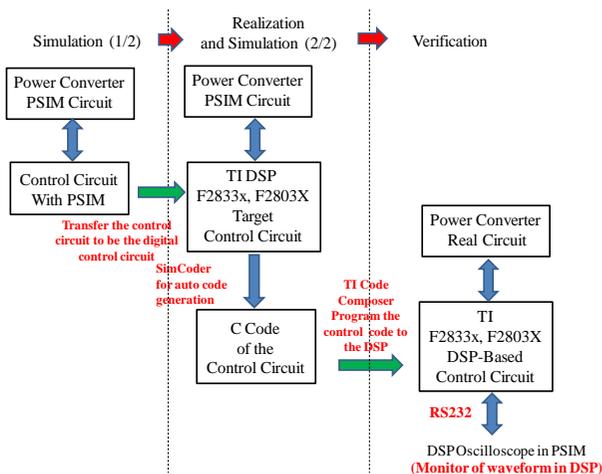
實驗 3 平均電流模式控制	97
實驗目的.....	97
實驗原理.....	97
電路模擬.....	108
SimCoder 程式規劃.....	109
實驗量測.....	110
實驗 4 最大功率點追蹤控制	111
實驗目的.....	111
實驗原理.....	111
電路模擬.....	125
SimCoder 程式規劃.....	129
實驗量測.....	138
實驗 5 PV 充電器	139
實驗目的.....	139
實驗原理.....	139
電路模擬.....	142
SimCoder 程式規劃.....	145
實驗量測.....	146
附錄 A SimCoder 概述	147
SimCoder 於模擬控制上設定.....	147
產生程式碼的元件 (Elements).....	150
附錄 B 產生程式碼 - 逐步介紹	151
連續模式系統.....	152
離散模式系統.....	153
產生硬體標的之程式碼.....	155
子電路 (Subcircuit) 產生程式碼.....	160
事件控制 (Event Control) 的系統.....	168
附錄 C 事件處理	171

基本概念.....	171
事件處理的元件.....	173
具有事件的子電路之限制.....	175
附錄 D SimCoder 資料庫.....	178
標準 PSIM 資料庫的元件.....	180
事件控制元件.....	184
全域變數.....	187
中斷.....	190
附錄 E F2833x 硬體標的.....	192
硬體配置.....	195
DSP 時脈.....	196
PWM 產生器.....	197
PWM 開始與停止.....	210
觸發區與其狀態.....	211
數位/類比轉換器.....	213
數位輸出與輸入.....	219
上/下計數器.....	221
編碼器與其狀態.....	239
捕捉與其狀態.....	242
串列傳輸界面(SCI).....	243
串列外設界面(SPI).....	246
專案設定與記憶體配置.....	254

簡介

電力轉換器採用數位控制是目前工業界產品的發展趨勢，數位控制可以提升電力轉換器的功能及性能，提高產品的附加價值，而且越來越多的電力轉換產品已開始採用數位控制技術。本教具的實施方式如圖 1.1 所示，目的在提供電力轉換器採用數位控制的學習平台，讓使用者透過 PSIM 軟體，除以模擬方式學習電力轉換器的原理、分析及設計外，亦可透過 PSIM 之 SimCoder 工具將控制電路轉換為數位控制程式，並可實際將以 DSP 取代之電路再作一次模擬，最後並可將透過模擬驗證過之控制程式燒錄於 DSP 晶片中，再透過 DSP 作控制及通訊，以驗證所設計電路及控制器之正確性。

圖 1.1
本教具使用程序



本教具的特點主要包含以下幾項

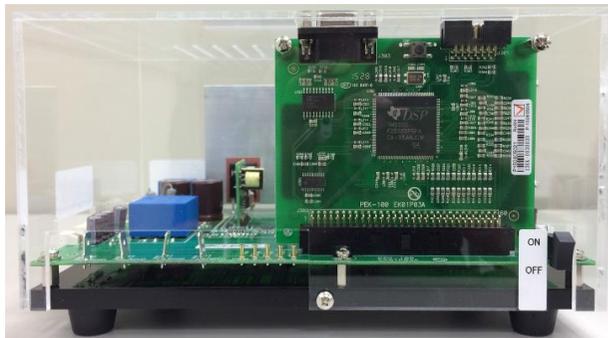
1. 可同時提供電力電子之分析、設計、模擬與實作驗證。
2. 在 PSIM 下以建立硬體電路的方式完成程式撰寫並燒錄程式，使完全不會 DSP 韌體撰寫的學習者亦能輕鬆完成程式撰寫，快速進入數位控制領域。
3. 本教具提供完備的實驗教材，包括 SimCoder 使用，以建立硬體方式撰寫程式的方法、詳細說明教具各部份電路，詳盡的實驗電路原理與設計，PSIM 電路模擬檔，DSP 硬體規劃及設定，程式燒錄方法等。
4. 本教具提供完整之教學投影片供教師及使用者參考。
5. 本教具後續若有增加實驗項目，免費提供之前購買者使用。

Buck DC-DC Converter 模組教具簡介

Buck DC-DC Converter 實驗模組如圖 1.2 所示，目前可提供以下 5 個實驗

1. 脈波寬度調變(Pulse Width Modulation, PWM)
2. 電壓模式控制(Voltage mode control)
3. 電流模式控制(Current mode control)
4. (1) 光伏最大功率點追蹤控制(PV MPPT control)-
擾動觀察法
(2) 光伏最大功率點追蹤控制(PV MPPT control)-
增量電導法
5. 光伏電池充電器(PV Battery charger)

圖 1.2
Buck DC-DC
Converter 實驗模
組



這些實驗模組除 Buck DC-DC Converter 主電力電路外尚包括以下組件

DSP 控制模組

- 提供 TI F28335 及 F28035 兩種模組。
- 各模組均具備隔離之 RS-232 通訊介面，可在實驗過程中將 DSP 內部信號傳回 PSIM 上觀測。

圖 1.3
DSP 控制模組



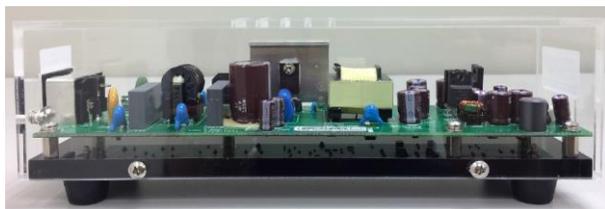
輔助電源模組

輸入電壓範圍為 100~250Vac 並提供+15V, -15V, 12V, 5V 等多組隔離電源輸出，總計最高 23W 之輸出，規格如表 1.1。

Description	Symbol	Min	Typ	Max	Units
Input					
Voltage	V _{IN}	100		250	VAC
Frequency	f _{LINE}	47	50/60	63	Hz
Output					
Output Voltage 1	V _{OUT1}	11.4	12	12.6	V
Output Current 1	I _{OUT1}	0.1	0.5	0.6	A
Output Voltage 2	V _{OUT2}	11.4	12	12.6	V
Output Current 2	I _{OUT2}	0.1	0.5	0.6	A
Output Voltage 3	V _{OUT3}	14.25	15	15.75	V
Output Current 3	I _{OUT3}	0.1	0.2	0.24	A
Output Voltage 4	V _{OUT4}	-14.25	-15	-15.75	V
Output Current 4	I _{OUT4}	-0.1	-0.2	-0.24	A
Output Voltage 5	V _{OUT5}	4.75	5	5.25	V
Output Current 5	I _{OUT5}	0.5	1	1.2	A
Total Output Power	P _{OUT}	7.505	23	28.98	W

表 1.1

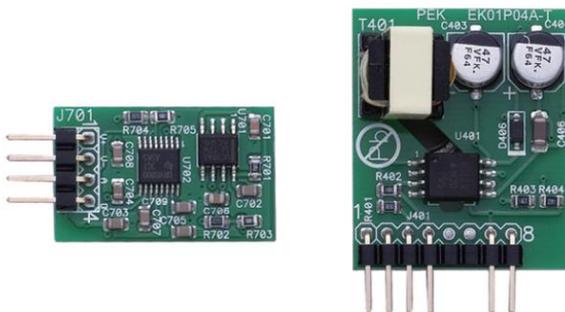
圖 1.4
Flyback 輔助電
源模組



驅動電源與開關驅動器模組

- 利用本模組只要一 12V 隔離電源便可提供任意多組輸出之隔離電源，解決複雜多組隔離驅動電源問題。
- 驅動器模組可提供高頻及高電流(2A)之驅動能力，並具備米勒效應之保護電路以避免誤動作發生。

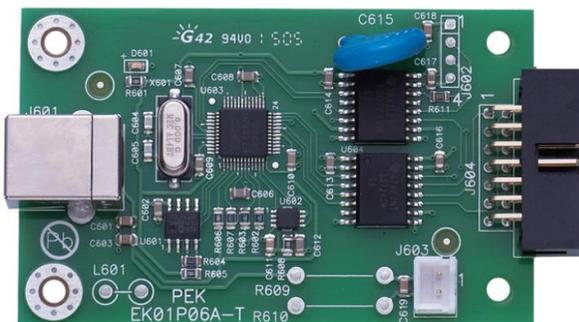
圖 1.5
驅動電源與開關
驅動器模組



JTAG 燒錄模組

提供隔離保護燒錄，避免因未隔離而在實驗過程導致電腦燒毀。

圖 1.6
JTAG 燒錄模組



教學實驗平台

固緯電子公司同時提供如圖 1.7 所示機架式之電力電子教學實驗平台以方便實驗以及教學之進行，教學平台包括：4CH 數位示波器(GDS-2304A(4ch, 300MHz))、兩 CH 直流電源供應器(PSW 250-4.5(250V, 4.5A), PSW 80-13.5(80V, 13.5A))、交流電源供應器(APS-7050 0~300V, 500W)、直流電子式負載(PEL-2000(2040: 350W, 80V))、交流功率表(GPM-8212)、交流被動式負載(GPL-100)等。電力電子教學實驗平台與單相 Inverter 模組教具配合之實驗方式如圖 1.8 所示，可以使用 PC 亦或 NB 作為 PSIM 模擬與本教具之燒錄與監控的實驗工具。



圖 1.7
機架式之電力電子教學實驗平台



圖 1.8
電力電子教學實驗平台與 Buck DC-DC Converter 模組教具配合之實驗方式

實驗目標

本教具以電路分析、設計、模擬、實驗等過程進行問題導向學習，根據轉換器規格進行量化設計其電力電路與控制器，並藉由 PSIM 模擬驗證，SimCoder 撰寫程式過程，使讀者徹底深入了解 Buck DC-DC Converter 之相關技術，培養以下能力

1. 電力轉換器之分析與設計能力。
2. PSIM 電路模擬能力。
3. 電力轉換器之控制器設計能力。
4. DSP 之數位控制技術(透過 SimCoder 輔助完成程式撰寫)。
5. 硬體與韌體之規劃及整合能力。
6. Step by step 完成電路製作與驗證能力。

本書章節說明

本書之章節安排如下

簡介	簡略介紹本教具之電路組成、實驗方式、實驗目標、本書各章之內容等。
PSIM 簡介	簡略介紹 PSIM 之組成與功能，讓使用者可以更加了解 PSIM 能夠協助轉換器電路分析與設計之工作內容。
教具硬體及設備說明	詳細介紹本教具中各個組成電路之工作原理與設備使用之方法。
實驗 1 脈波寬度調變	學習脈波寬度調變(PWM)之原理、Buck DC-DC 模組之導通模式、開迴路電壓及電流量測方法，TI F28335 DSP IC 腳位設定、DSP 之 PWM 及 A/D 模組設定、RS232 監控 DSP 內部信號之方法等。
實驗 2 電壓模式控制	學習 Buck converter 之小信號模型推導方法、AC sweep 掃波德圖方法，電壓迴路控制器設計方法、硬體規劃及 SimCoder 程式撰寫等。
實驗 3 電流模式控制	學習 Buck converter 之平均電流模式控制方法，包括電流迴路及電壓迴路之模型推導，控制器設計、硬體規劃及 SimCoder 程式撰寫等。
實驗 4 光伏(PV)模組最大功率點追蹤(MPPT)控制	學習 PV 模組之最大功率點追蹤控制方法、包括擾動觀察法、增量電導法等常用之 MPPT 控制方法、硬體規劃及其控制程式之撰寫等。
實驗 5 PV 充電器	學習 PV 充電器之控制方法、包括電池三階段充電控制方法、MPPT 控制與充電控制方法之結合、硬體規劃及其控制程式之撰寫等。

附錄 A SimCoder 詳細介紹 PSIM 之 SimCoder 以及 TI F28335
概述 Target 之使用方法。

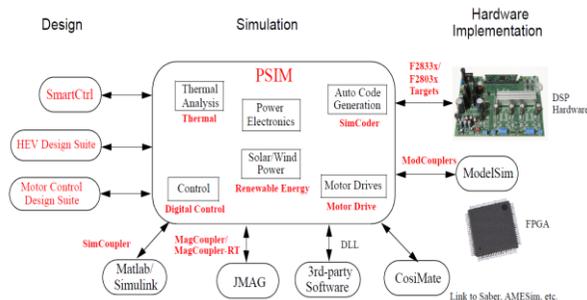
PSIM 簡介

PSIM 為一專門為各式電力電子、馬達驅動及電力轉換等系統所設計之模擬軟體，其特點為：功能全面、元件完整、模擬速度快、模擬結果精確、非常容易使用等，為目前國際上學術界及業界常用之教學與研究軟體，因此本書藉此軟體當成平台，使讀者能與國際之研究及教學同步。

PSIM 具有模擬、設計及硬體電路實現等全方位能力，提供之功能如圖 2.1 所示，除了其主體可以提供電力電子電路模擬外，亦包含了以下模組

圖 2.1

PSIM 提供之模擬功能



Motor Driver 模組

包含了直流馬達、直流無刷馬達、鼠籠式感應馬達、繞線式感應馬達、永磁同步馬達、同步馬達、開關磁阻馬達及各式速度、位置、轉矩迴授裝置、各式機械負載及驅動源裝置等，可以作各式馬達及發電機等應用系統之模擬。

Digital Control 模組

包含了各式離散元件如 zero-order hold, z-domain 轉移函數、數位濾波器、quantization blocks 等可執行數位控制及分析的元件。

SimCoupler 模組	用以作為 PSIM 與 Simulink 之介面，使 PSIM 與 Simulink 可作 Co-simulation(同時模擬)，讓原先使用 Simulink 的使用者亦可使用原先發展的技術，因使用 PSIM 改善其模擬速度及收斂性，亦讓原 PSIM 的使用者可以透過 Simulink 使用 Matlab 豐富的 Toolbox 功能。
Thermal 模組	提供實際功率半導體元件特性的模型，可以計算功率半體元件的損失，作為計算溫升及設計散熱機構參考，使用者亦可以根據實際元件資料手冊自己建立考慮 Thermal 的功率半導體元件。
Renewable Energy 模組	包含太陽光電(PV)模組、風機(wind turbine)及電池模型。
SimCoder 模組	可以將控制電路自動轉化為 C 程式，並可透過 TI Composer 對 DSP 晶片作燒錄，透過 PSIM 提供一個硬體與軟體人員可以彼此討論的平台，建立更緊密協同開發的環境。
F2833X Target	包含一些 TI DSP F2833X 之元件資料庫，可以自動產生燒錄 F2833X 的程式。
F2803X Target	包含一些 TI DSP F2803X 之元件資料庫，可以自動產生燒錄 F2803X 的程式。
MagCoupler 模組	提供 PSIM 與磁路分析軟體 JMAG 之介面，使二者可作 Co-simulation。
MagCoupler-RT 模組	提供 PSIM 與磁路分析軟體 JMAG 資料檔之連結。
ModCoupler 模組	提供 PSIM 與 ModelSim 之介面，使二者可作 Co-simulation。其包括兩個版本，ModCoupler-VHDL 支援 VHDL 程式以及 ModCoupler-Verilog 支援 Verilog 程式

HEV Design Suite	預設一些設計樣版(template)提供混合電動車(HEV)之傳動系統(powertrain system)輔助設計
------------------	--

Motor Control Design Suite	預設一些設計樣版(template)提供感應馬達、線性及非線性永磁同步馬達驅動器的輔助設計
----------------------------	---

除此 PSIM 亦提供與 CosiMate 之連結，藉由 CosiMate 平台可與許多軟體作 co-simulation，包括 Matlab/Simulink, ModelSim, Saber (from Synopsys), Easy5 and Adams (from MSCSoftware), Inventor (from Autodesk), AMESim (from LMS), GT-Power (from Gamma Technologies) 等等，細節請連結 www.chiastek.com。

教具硬體及設備說明

本教學實驗教具為降壓式轉換器，輸入設備為 GW PSW 160-7.2 360W，輸入電壓操作範圍為 30~70V，輸出設備為 GW PEL-2004 與 PEL-2040，示波器為 GW GDS-2304A,這些設備在本章會概略介紹。

功率級電路

本實驗教具之實體如圖 3.1，電路圖為圖 3.2，輸入電壓因安全考量設定在 50V，輸出電壓為 24V。輸入端先經過一個 10A 的保險絲，接著並聯兩個 100 μ F/250V 輸入電解電容，隨後一個由 MOS 與二極體及電感(365 μ H)組成的降壓式轉換器，MOS 的驅動電路將在後面說明，後端為三個 100 μ F/250V 的輸出電解電容並聯，最後接至輸出端。

圖 3.1

Buck DC-DC
Converter 實驗模
組



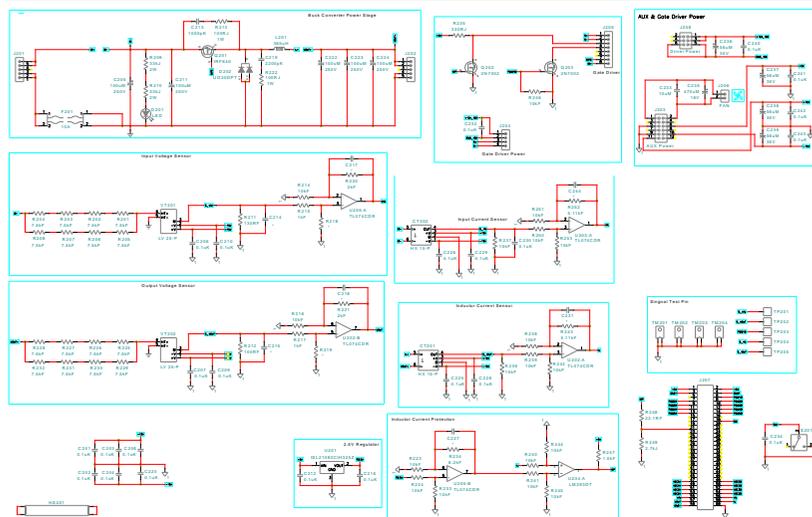


圖 3.2 Buck DC-DC Converter 電路圖

在控制及回授電路方面，輸入電壓(V_{IN})，輸出電壓(V_0)，輸入電流(I_{IN}) 與輸出電流(I_0) 等四個參數經過取樣後送入 DSP，以下分別說明這四個參數送入 DSP 前的取樣過程。

1. 由圖 3.2 摘錄輸入電壓的取樣過程，如圖 3.3，經 8 顆電阻衰減 1/15k 倍送入型號為 LEM LV 25-P 的取樣 IC，經 IC 的轉換比例放大 2.5 倍後，再乘上 100Ω 後得到 V_{IN} 信號，再經倍率 1.44 的 OP 放大器後送入 DSP，取樣衰減的比例如(3.1)所示。

$$Gain = \frac{1}{15k} \times 2.5 \times 100 \times 1.44 = 0.024 \quad (3.1)$$

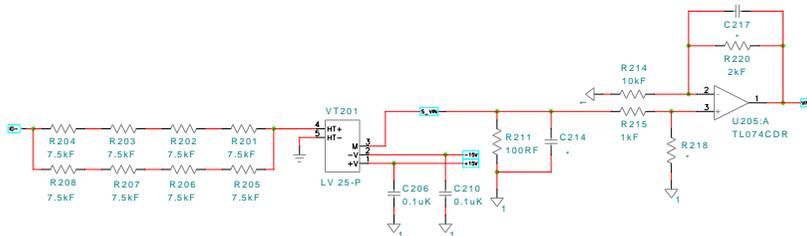


圖 3.3 輸入電壓取樣電路

- 由圖 3.2 摘錄輸出電壓的取樣過程，如圖 3.4，經 8 顆電阻衰減 1/15k 倍送入型號為 LEM LV 25-P 的取樣 IC，經 IC 的轉換比例放大 2.5 倍後，再乘上 100Ω 後，得到 V_0 信號，再經倍率 1.44 的 OP 放大器後送入 DSP，取樣衰減的比例如(3.2)所示。

$$Gain = \frac{1}{15k} \times 2.5 \times 100 \times 1.44 = 0.024 \quad (3.2)$$

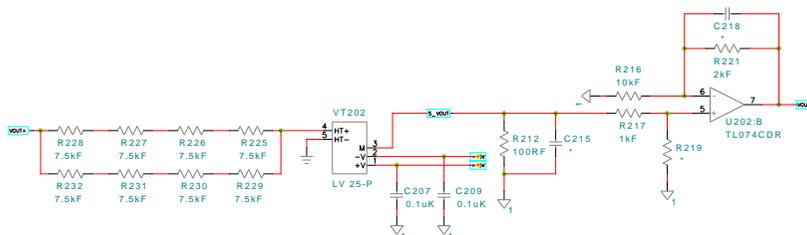


圖 3.4 輸出電壓取樣電路

- 由圖 3.2 摘錄輸入電流的取樣過程，如圖 3.5，經電流感測 IC LEM HX 15-P，其轉換比例為 2.67×10^{-5} 倍，再乘上 10kΩ，再經倍率 1.125 的 OP 放大器後送入 DSP，取樣衰減的比例如(3.3)所示。

$$Gain = (2.67 \times 10^{-5}) \times 10k \times 1.125 = 0.3 \quad (3.3)$$

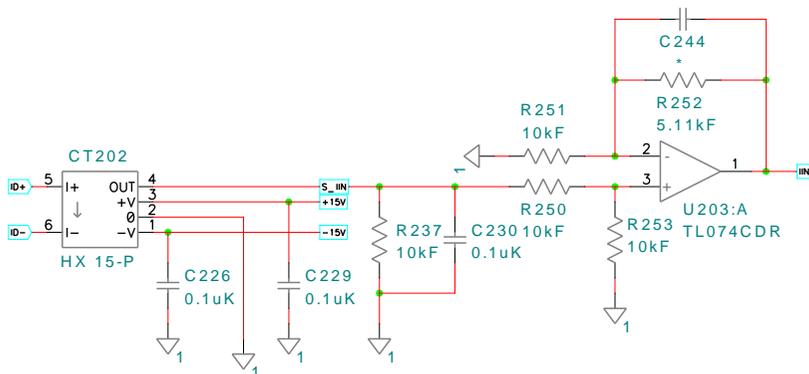


圖 3.5 輸入電流取樣電路

- 由圖 3.2 摘錄電感電流的取樣過程，如圖 3.6，經電流感測 IC LEM HX 15-P，其轉換比例為 2.67×10^{-5} 倍，再乘上 10kΩ，再經倍率 1.125 的 OP 放大器後送入 DSP，取樣衰減的比例如(3.4)所示。

$$Gain = (2.67 \times 10^{-5}) \times 10k \times 1.125 = 0.3 \quad (3.4)$$

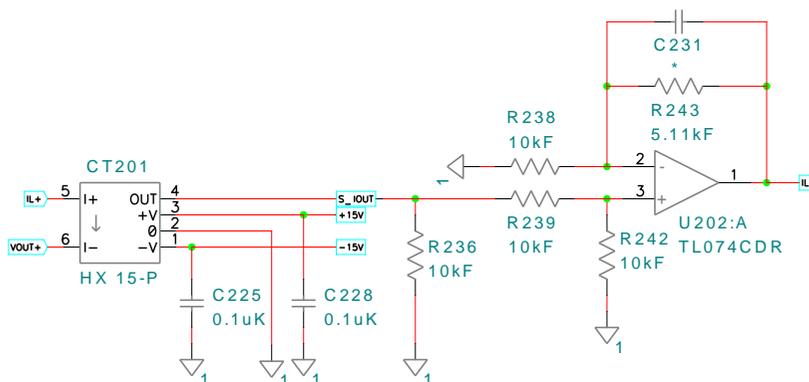


圖 3.6 電感電流取樣電路

硬體電路有設計過電流保護，常態為高準位，一旦電流過高則輸出變為低準位，觸發保護使轉換器停止運作以免損壞。電路如圖 3.7。

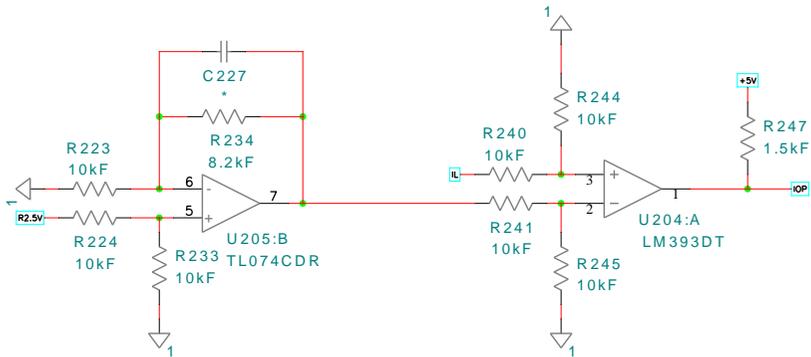


圖 3.7 過電流保護電路

降壓式轉換器教具提供下列測試點供使用者量測：

1. DSP 送出 PWM 再經過驅動電路進而驅動 MOS，PWM 信號可由量測點觀測。
2. 輸入電壓(V_{IN})經 8 顆電阻衰減 1/15k 倍送入取樣 IC，經 IC 的轉換比例放大 2.5 倍，再乘上 100 Ω 後，即為量測點的觀測值，衰減倍率如(3.5)：

$$Gain = \frac{1}{15k} \times 2.5 \times 100 = 1.67 \times 10^{-2} \quad (3.5)$$

3. 電感電流(V_0) 經 8 顆電阻衰減 1/15k 倍送入取樣 IC，經 IC 的轉換比例放大 2.5 倍，再乘上 100 Ω 後，即為量測點的觀測值，衰減倍率如(3.6)：

$$Gain = \frac{1}{15k} \times 2.5 \times 100 = 1.67 \times 10^{-2} \quad (3.6)$$

4. 輸出電流(I_{IN})轉換比例為 2.67×10^{-5} 倍的電流感測器，再乘上 10k Ω ，即為量測點的觀察值，衰減倍率如(3.7)：

$$Gain = (2.67 \times 10^{-5}) \times 10k = 0.267 \quad (3.7)$$

5. 輸出電壓(I_0) 轉換比例為 2.67×10^{-5} 倍的電流感測器，再乘上 10k Ω ，即為量測點的觀察值，衰減倍率如(3.8)：

$$Gain = (2.67 \times 10^{-5}) \times 10k = 0.267 \quad (3.8)$$

DSP 控制電路

DSP 控制電路以 TI TMS320F28335 實現之硬體如圖 3.8，電路圖為圖 3.9，由雙輸出穩壓 IC 提供 3.3V 與 1.8V，3.3V 即為 28335 IC 的工作電源。信號送入 DSP 前會先經過二極體箝位電路以確保進入 IC 之電壓都在 0~3V 之間避免損壞 DSP。透過隔離的 RS232 通信介面,在實驗中可即時將 DSP 內部信號傳回 PSIM 中的示波器觀測。F28335 控制板各腳位輸出定義如下表

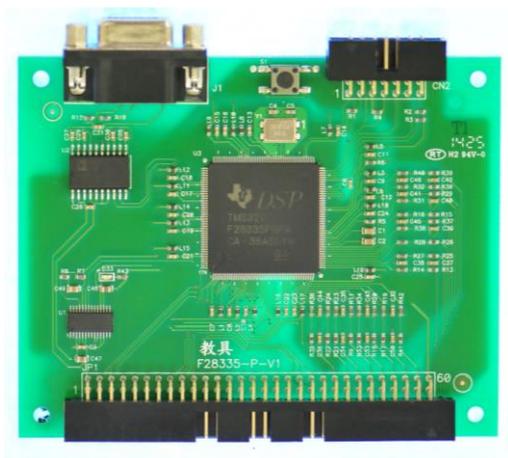
	Pin	
+5V in	1	2 +5V in
GND	3	4 GND
GPIO-00 / EPWM-1A	5	6 GPIO-01 / EPWM-1B / MFSR-B
GPIO-02 / EPWM-2A	7	8 GPIO-03 / EPWM-2B / MCLKR-B
GPIO-04 / EPWM-3A	9	10 GPIO-05 / EPWM-3B / MFSR-A / ECAP-1
GPIO-06 / EPWM-4A / SYNCI / SYNCO	11	12 GPIO-07 / EPWM-4B / MCLKR-A / ECAP-2
GPIO-08 / EPWM-5A / CANTX-B / ADCSOC-A	13	14 GPIO-09 / EPWM-5B / SCITX-B / ECAP-3
GPIO-10 / EPWM-6A / CANRX-B / ADCSOC-B	15	16 GPIO-11 / EPWM-6B / SCIRX-B / ECAP-4
GPIO-48 / ECAP5 / XD31 (EMIF)	17	18 GPIO-49 / ECAP6 / XD30 (EMIF)
GPIO-50	19	20 GPIO-51
GPIO-12 / TZ1n / CANTX-B / MDX-B	21	22 GPIO-13 / TZ2n / CANRX-B / MDR-B
GPIO-15 / TZ4n / SCIRX-B / MFSX-B	23	24 GPIO-14 / TZ3n / SCITX-B / MCLKX-B
GPIO-24 / ECAP1 / EQEPA-2 / MDX-B	25	26 GPIO-25 / ECAP2 / EQEPB-2 / MDR-B
GPIO-26 / ECAP3 / EQEPI-2 / MCLKX-B	27	28 GPIO-27 / ECAP4 / EQEPS-2 / MFSX-B
GPIO-16 / SPISIMO-A / CANTX-B / TZ-5	29	30 GPIO-17 / SPISOMI-A / CANRX-B / TZ-6
GPIO-18 / SPICLK-A /	31	32 GPIO-19 / SPISTE-A / SCIRX-B

SCITX-B

GPIO-20 / EQEP1A / MDX-A / CANTX-B	33	34	GPIO-21/ EQEP1B/ MDR-A/ CANRX-B
GPIO-22 / EQEP1S / MCLKX-A / SCITX-B	35	36	GPIO-23/ EQEP1I/ MFSX-A / SCIRX-B
GPIO-28 / SCIRX-A / -- / TZ5	37	38	GPIO-29 / SCITX-A / -- / TZ6
GPIO-30 / CANRX-A	39	40	GPIO-31 / CANTX-A
GPIO-32 / I2CSDA / SYNCI / ADCSOCA	41	42	GPIO-33 / I2CSCL / SYNCO / ADCSOCB
ADCIN-B7	43	44	ADCIN-A7
ADCIN-B6	45	46	ADCIN-A6
ADCIN-B5	47	48	ADCIN-A5
ADCIN-B4	49	50	ADCIN-A4
ADCIN-B3	51	52	ADCIN-A3
ADCIN-B2	53	54	ADCIN-A2
ADCIN-B1	55	56	ADCIN-A1
ADCIN-B0	57	58	ADCIN-A0
GND	59	60	GND

圖 3.8

DSP 控制電路



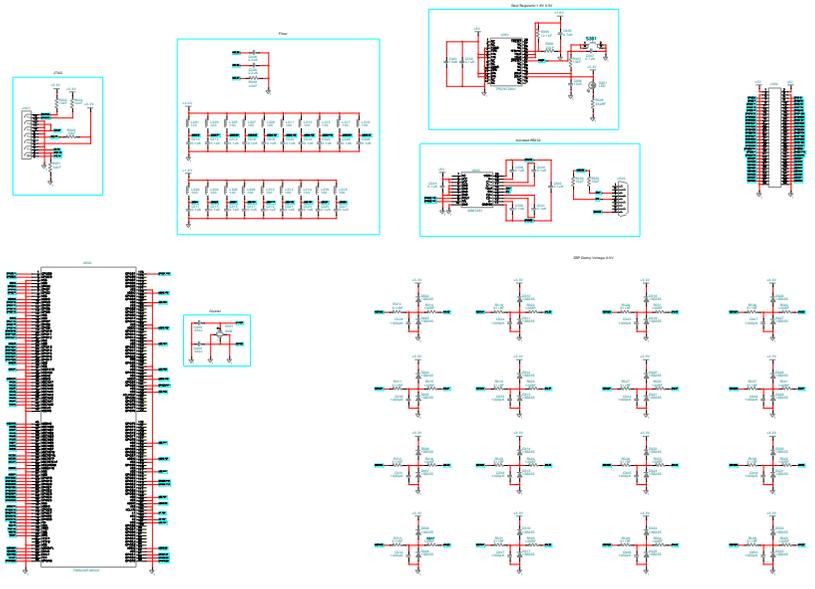
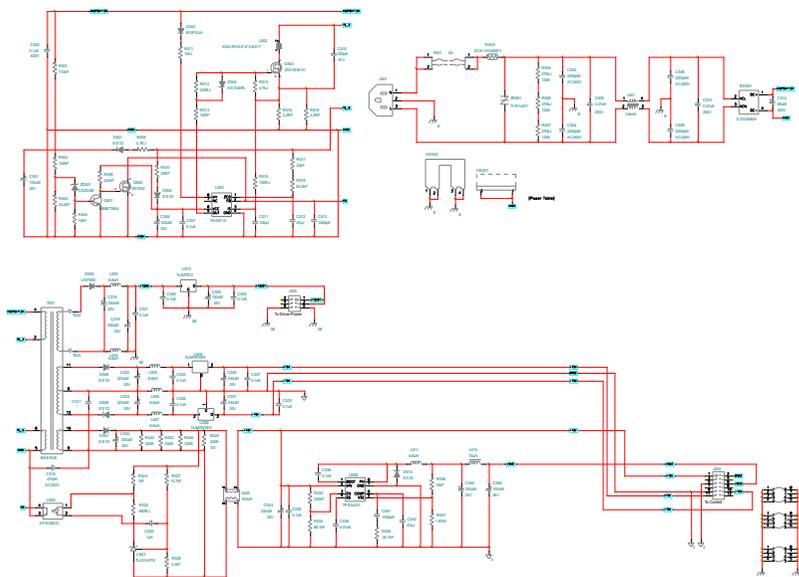


圖 3.9 F28335 電路圖

輔助電源

此模組以 Flyback 為設計架構，輸入電壓範圍為 100~250V，輸出為三組不共地的隔離電源，分別是 (1)12V (2)12V, 5V (3)15V, -15V 實體為圖 3.10，電路如圖 3.11

圖 3.10
輔助電源



3.11 輔助電源電路

驅動電路

驅動電源模組由 Gate Driver 板與 Gate Driver Power 板組成，可提供多組隔離電源，圖 3.12 左為 Gate Driver，右為 Gate Driver Power，圖 3.13, 3.14 為電路圖。輸入一個 12V 電壓至 Gate Driver Power，其輸出為±12V 之方塊波。Gate Driver 的輸入為此±12V 之方塊波與由 DSP 產生的 PWM 信號，輸出為驅動 MOS 的信號。Gate Driver 藉由變壓器與光耦合驅動 IC 達到隔離的目的。

圖 3.12

電路驅動電路模組

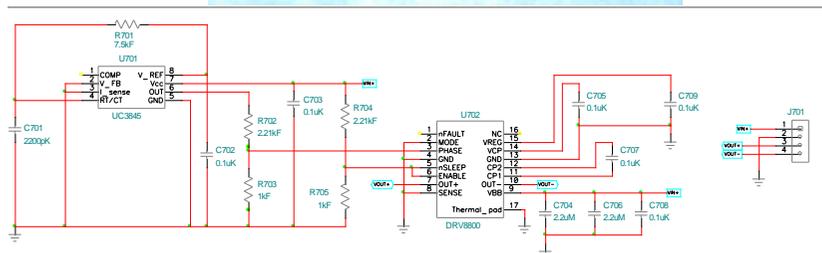
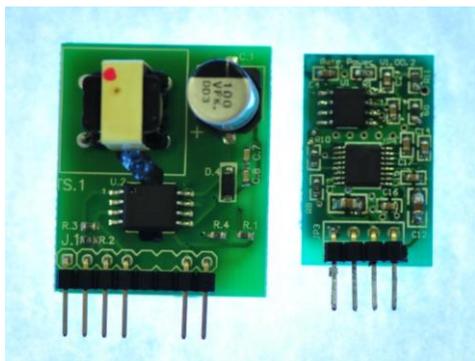


圖 3.13 Gate Driver Power 電路圖

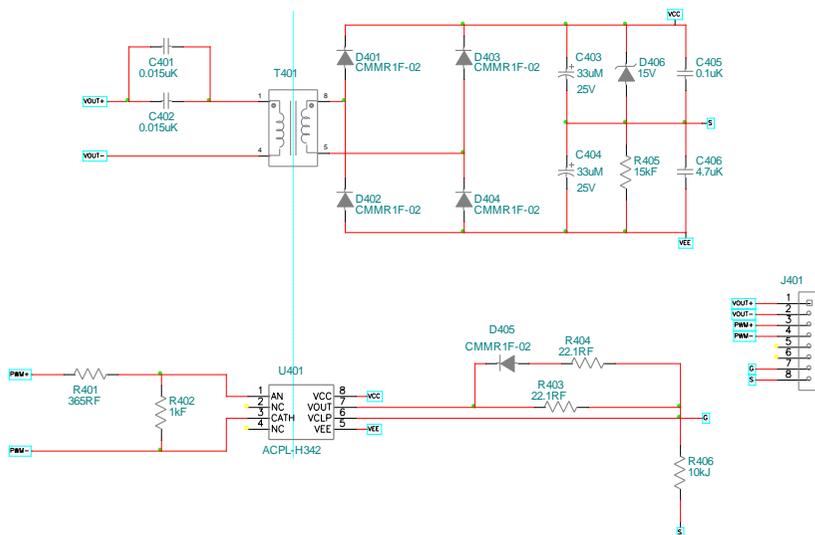


圖 3.14 Gate Driver 電路圖

JTAG 燒錄電路

此模組可將程式碼由電腦端燒錄至 DSP 晶片，硬體電路如圖 3.15，電路圖為圖 3.16，以 USB 與電腦端連接，JTAG 與 DSP 端連接。

圖 3.15

USB_JTAG 燒錄
電路

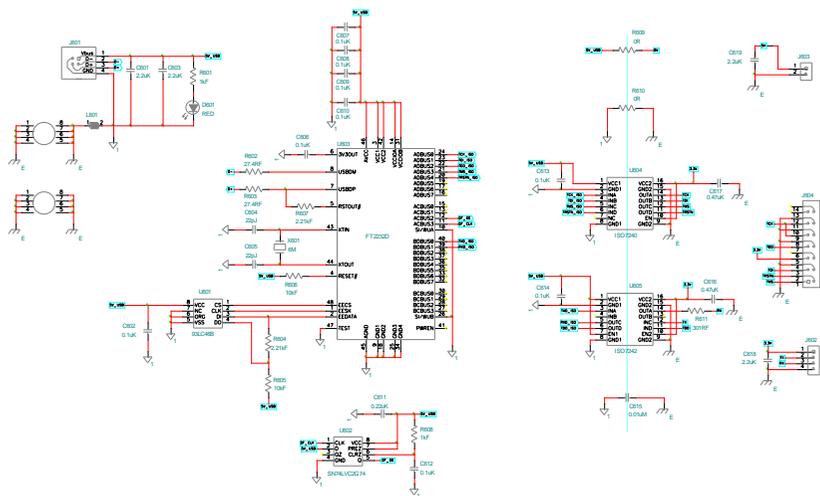


圖 3.16 USB_JTAG 電路

實驗設備說明

PSW 250-4.5

輸入使用 GW PSW 250-4.5 360W 可程式交換直流電源供應器，額定電壓輸入為 250V，輸出功率 360W，如圖 3.17。可定功率多段範圍操作在額定功率下，儀器特點說明如下

- 定功率多段(V&I)範圍操作
- CC/CV 優先模式選擇,適用於電池及 LED 產業
- 可調式 Slew Rate
- 最多可兩台串聯操作(僅 30V,80V,160V 機種)
- 最多可三台並聯操作
- 高效能及高功率密度電源
- 標準通訊介面:LAN, USB,外部類比遠端控制端子
- 選配通訊介面:GPIB-USB 調適器
- 支援 LabVIEW Driver

圖 3.17

PSW 250-4.5
360W



APS-7050

當進行市電並聯實驗中為了安全性考量，不直接與市電並聯，而是以可程式交流電源供應器 GW APS-7050 模擬市電，如圖 3.18，儀器有下列特點

- 低輸出連波與雜訊
- 優良且功能豐富的量測能力
- 標準最大輸出電壓為 310Vrms
- 最大輸出電壓和頻率 600Vrms/999.9Hz

圖 3.18
APS-7050



GDS-2304A

量測波形信號時使用 GDS-2304A 300MHz，4 通道，彩色數位儲存示波器，如圖 3.19，特點如下

- 300MHz 頻寬
- 2GSa/s 即時取樣率與 100GSa/s 等效取樣率
- 2M 記憶體深度
- 標準內建波形搜尋及分段記憶體功能
- 最高每秒 80,000 次波形更新
- 標準提供 UART ,I2C 與 SPI 觸發與解碼功能
- 可選配 8 或 16 數位通道以升級為混合式示波器
- 數位通道標準包含 UART ,I2C 與 SPI 串列與並列 (parallel) 匯流排觸發與解碼
- 可選配 5MHz 及 25MHz 訊號產生器
- 彈性介面選擇(標準:USB ;選配:LAN/GPIB)
- 可免費升級 CAN/LIN 匯流排分析,進階邏輯,資料紀錄器,數位電壓錶及水平觸發定點展開功

圖 3.19
GDS-2304A



PEL-2000

直流負載使用 PEL-2040 與 PEL-2004，如圖 3.20，特點如下

- 300MHz 頻寬 10 組立即鍵
- 120 組呼叫/儲存記憶裝置
- 並聯操作增加此用功率
- 外部波形控制功能
- 序列編輯功能模擬負載實際狀況
- 過功率、過電流、過電壓、低電壓保護，所有保護準位皆可調
- 多重使用者介面:USB Device/Host,RS-232,GPIB

圖 3.20
PEL-2000



GPM-8212

需要量測輸出電壓、電流與功率時，以 GPM-8212 交流功率計量測，如圖 3.21，其儀器特點如下

- 同時量測顯示功率、電流、電壓 (功率因素或頻率)
- True RMS 電壓、電流、功率
- 最大值 / 最小值 / 保存功能
- 高雜訊抗拒能力
- CT 比率設定 (1 ~ 9999)
- PT 比率設定 (1 ~ 9999)

圖 3.21
GPM-8212



GPL-100

此模組為被動式負載提供整流性負載與電阻性負載，並可恣意地切換，負載功率可做三段調整，如圖 3.22，特點整理如下

- 可切換整流性負載與電阻性負載
- 負載功率可三段切換，分別為 1/3 載，2/3 載以及滿載
- 三段電阻各為 42 歐姆
- 整流性負載 PF 值為 0.8

圖 3.22
GW GPL-100



實驗 1 脈波寬度調變 (PWM)

實驗目的

- 學習 PWM 切換之原理
- Buck 工作模式
- 開迴路電壓及電流量測方法
- TI F28335 DSP IC 腳位設定
- DSP 之 PWM 及 A/D 模組設定
- RS232 監控 DSP 內部信號之方法
- 熟悉本實驗之硬體電路操作

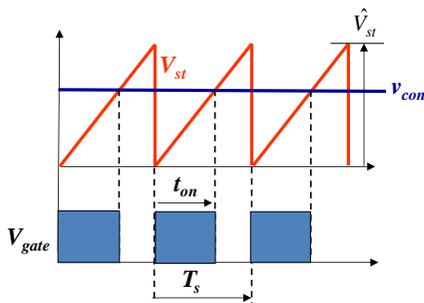
實驗原理

PWM (Pulse Width Modulation, 脈波寬度調變)原理

PWM 乃切換式轉換器將控制信號轉換成開關 ON/OFF 觸發信號方法中最典型的方法，如圖 4.1 所示，鋸齒波頻率(f_s)即為開關之切換頻率，切換頻率之倒數稱為切換週期($T_s=1/f_s$)。開關導通時間(T_{on})與切換週期(T_s)之比值稱為開關之責任週期(D , duty ratio or duty cycle)， D 與 PWM 控制電壓 V_{con} 及鋸齒波 V_{st} 之關係滿足：

$$D = \frac{t_{on}}{T_s} = \frac{v_{con}}{\hat{V}_{st}} \quad (4.1)$$

圖 4.1
PWM 波形



其中 \hat{V}_{st} 為鋸齒波之振幅(peak-to-peak)，鋸齒波一般為固定頻率，但為提高輕載時的效率，亦或降低空載時的損耗，許多轉換器會使切換頻率隨負載降低，以降低在輕載下主要的損耗-切換損失。另外為減輕電磁干擾(EMI)問題，一般控制 IC 在定頻下亦會作抖頻(jitter)，即切換頻率為 $f_s \pm f_d$ ， f_d 為抖頻範圍。如此切換能量可以分散成一頻帶而不會集中在一頻率上。抖頻 f_d 一般遠低於切換頻率，例如 65kHz 的轉換器其抖頻在幾 kHz(例如 5kHz)範圍，而掃頻之週期在幾個 ms 範圍。

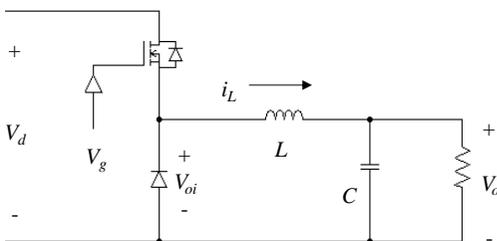
切換頻率之影響

切換頻率是轉換器特性的根本，因其決定轉換器之體積、重量、響應速度、效率等特性，圖 4.2 可用以說明切換頻率對於體積之影響。在電感電流 i_L 為連續導通模式下(即 $i_L > 0$)，當開關導通時 V_{oi} 電壓為輸入電壓 V_d ，電感電流開始流通，當開關截止時電感電流經由二極體持續導通， V_{oi} 電壓為零，因此圖 4.2(b)中 V_{oi} 電壓為一與責任週期相同波形之方塊波。週期性的 V_{oi} 波形可以藉由傅立葉級數(Fourier Series)分解成直流、基本波(即切換頻率)、以及切換頻率整倍數之諧波，其頻譜如圖 4.2(b)所示。 LC 在此扮演低通濾波器的角色，輸出電壓 V_o 與 V_{oi} 電壓之轉移函數為：

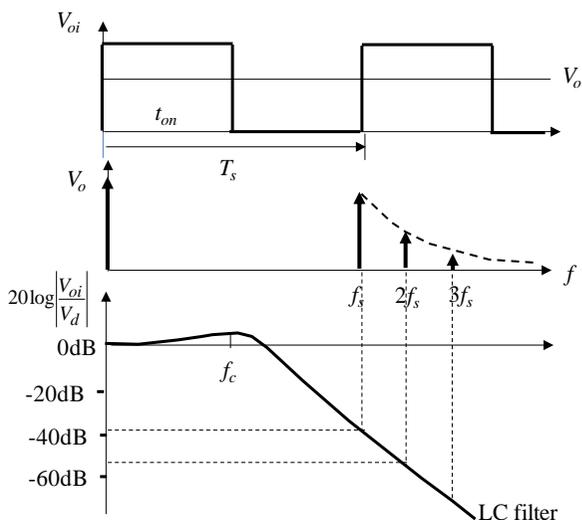
$$H(s) = \frac{V_o(s)}{V_{oi}(s)} = \frac{R // \frac{1}{sC}}{sL + (R // \frac{1}{sC})} = \frac{1}{s^2 + R/Ls + 1/LC} \quad (4.2)$$

LC 低通濾波器之波德圖如圖 4.2(b)所示，其截止頻率 $f_c = (1/(2\pi\sqrt{LC}))$ 遠低於切換頻率， V_{oi} 經過 LC 濾波器後，除其直流成份外均會被低通濾波器以 -40dB/decade 的速度衰減，最終使得 V_o 之基本波及各次諧波項成份幾乎可被忽略，使其波形為接近無漣波之純直流。圖 4.2(b) 可說明當切換頻率愈高，欲獲得同樣之高頻切換成份衰減， LC 之截止頻率 f_c 亦可以相對提高，因此 L 與 C 之值便可以相對減少。由於轉換器主要的體積來源在於這些 LC 被動式元件以及功率半導體元件， LC 值減小 L 與 C 的體積便可縮小，提高切換頻率是縮小轉換器體積最有效的方法。然而提高切換頻率需克服散熱(Thermal issue)以及 EMI 的問題，提高切換頻率後功率半導體元件的切換損失亦會增加，為處理散熱問題需要增加散熱面積以降低熱阻，例如增加散熱片大小、加裝風扇等。因此提高切換頻率以降低被動式元件大小會受到增加散熱體積的障礙。其次提高切換頻率亦將提高電路之 dv/dt 及 di/dt 程度，造成在更小體積內處理更劇烈 EMI 問題的困難度。

圖 4.2
 切換頻率與低通
 濾波器之關係說
 明：
 (a)降壓式轉換器



(b) V_{oi} 波形及其
 頻譜



如上所述，提高切換頻率亦會增加半導體功率元件之切換損失，亦會提高磁性元件之鐵損，然而由於體積降低可以減少線路長度之損耗，亦使用較低雜散損耗之元件，因此提高切換頻率對負載在輕載或重載的影響不一，需視實際電路個別情況而定。就電壓及電流之響應速度而言，提高切換頻率降低被動式元件值通常可以提高系統開迴路控制的頻寬，間接增加閉迴路下系統的響應速度。

降壓式轉換器(Buck Converter)工作原理

降壓式轉換器電路相關參數如圖 4.3 所示，依據輸出電感電流導通的方式分類可分成連續導通模式(CCM, continuous conduction mode)、邊界導通模式(BCM, boundary conduction mode)與不連續導通模式(DCM, discontinuous conduction mode)。

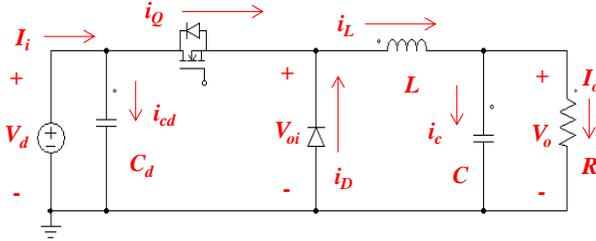


圖 4.3 降壓式轉換器電路相關參數表示

A. 連續導通模式(CCM)

連續導通模式即電感電流一週期內的波形均在 0 之上，操作在 CCM 的降壓式轉換器可分成開關 ON 與開關 OFF 兩個狀態如圖 4.4 所示，所對應的工作波形則如圖 4.5 所示。當開關(Q)ON 時(狀態①)，二極體(D)反向偏壓截，能量由輸入直接傳至輸出側，電感電壓為：

$$V_L = V_d - V_o \tag{4.3}$$

由於 \$V_L\$ 值大於零，因此電感電流 \$i_L\$ 為上升(上升斜率為 \$di_L/dt = V_L/L\$)，穩態下開關導通期間 \$i_L\$ 由 \$I_1\$ 上升至 \$I_2\$。

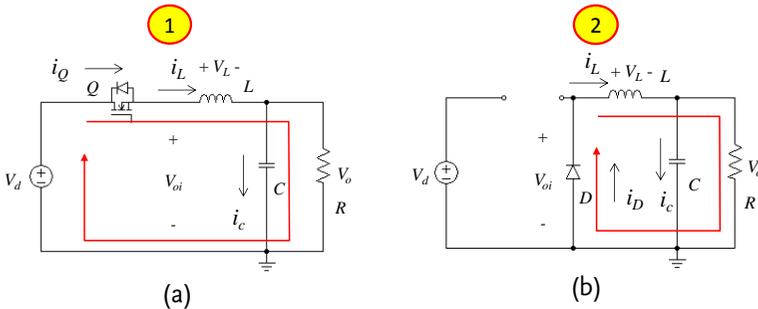


圖 4.4 降壓式轉換器於 CCM 下之工作狀態：(a)開關 ON(b)開關 OFF

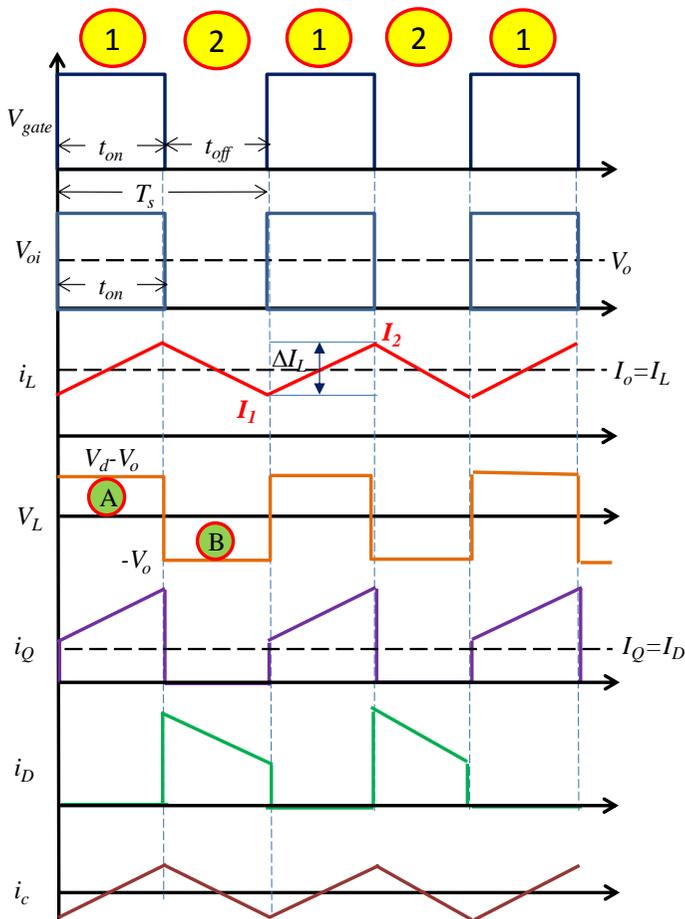


圖 4.5 降壓式轉換器於 CCM 下之工作波形

當開關 OFF 時(狀態②)，二極體導通，電感能量巡迴於二極體及輸出，電感電壓為：

$$V_L = -V_o \quad (4.4)$$

由於 V_L 值小於零，因此電感電流 i_L 為下降(下降斜率為 $di_L/dt = -V_o/L$)，穩態下二極體導通期間 i_L 由 I_2 下降至 I_1 。電感之 B-H(磁通密度-磁場強度)曲線如圖 2.6 所示，B-H 曲線相當於法拉第定律與安培定律之展現，對電感而言， N 匝線圈流過電流 I 將造成磁動

勢(mmf)：

$$F = NI = Hl \tag{4.5}$$

其中 $l(m)$ 為為電感之導磁路徑，由(4.5)可知磁場強度(H)與流通之電流 $I(A)$ 成正比，因此 H 之單位可表示為 (A/m) 。當降壓式轉換器開關 ON 時(狀態①)電感電流由 I_1 上升至 I_2 ，其在 $B-H$ 曲線之工作如圖 4.6 所示，將沿著 $B-H$ 曲線之下緣由 B_1 變化到 B_2 ；當降壓式轉換器開關 OFF 時(狀態②)電感電流將由 I_2 下降回到 I_1 ，其在 $B-H$ 曲線之工作則將沿著 $B-H$ 曲線之上緣由 B_2 回到 B_1 。 $B-H$ 曲線與 B 軸所夾之面積為此時電感所儲存之能量，因此 ON 及 OFF 一次將有一由 $B-H$ 曲線工作迴圈所圍成面積(即能量)之損耗，此能量乘以切換頻率(f_s)或此能量除以切換周期(T_s)可得到一功率(power)損耗稱為鐵損(core loss)，切換頻率越高鐵損越嚴重。由法拉第定律可知：

$$N\Delta BA = N\Delta\phi = \Delta\int Edt \tag{4.6}$$

其中 $A(m^2)$ 為鐵心之截面積， ϕ 為磁通量(Wb)， B 為磁通密度 $(Wb/m^2=Tesla, 1 Tesla=10000Gauss)$ ， E 為電感之電壓(V)。

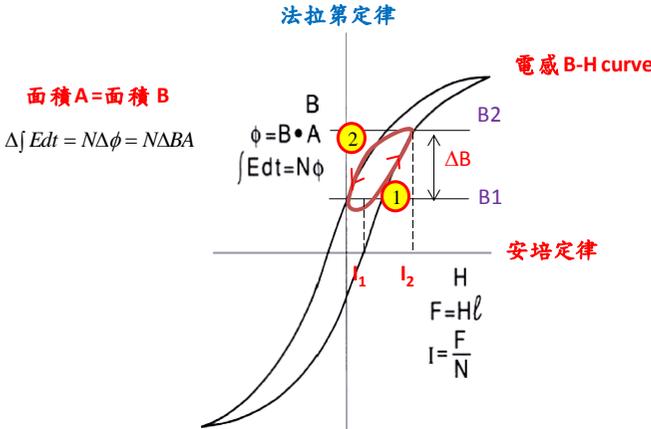


圖 4.6 電感之 $B-H$ 曲線及工作迴圈

穩態下開關 ON 與開關 OFF 之 ΔB (磁通密度變化)相同，由(4.6)可知

$$\Delta \int_{ON} Edt = \Delta \int_{OFF} Edt \tag{4.7}$$

亦即開關 ON 與開關 OFF 期間電感電壓對時間之積分相等(亦可解釋為電感電壓一週期之平均值為零)，因此圖 4.6 之電感電壓波形中，面積 A=面積 B，此稱為伏秒平衡(voltage-second balance)。利用圖 4.6 之面積 A=面積 B 可得：

$$(V_d - V_o)t_{on} = V_o t_{off} \quad (4.8)$$

利用(4.8)整理可得操作在 CCM 之降壓式轉換器的電壓轉換比為：

$$\frac{V_o}{V_d} = D \quad (4.9)$$

假設降壓式轉換器的效率為 100%，則 $P_o = P_d$ ，因此

$$V_o I_o = V_d I_i \quad (4.10)$$

又

$$I_o = I_L = \frac{V_o}{R} \quad (4.11)$$

故

$$I_i = I_Q = D I_o \quad (4.12)$$

$$I_D = (1 - D) I_o \quad (4.13)$$

由圖 4.6 之波形可得：

$$\Delta I_L = I_2 - I_1 = \frac{V_d - V_o}{L} t_{on} = \frac{V_d}{L} (1 - D) D T_s = \frac{V_o}{L} (1 - D) T_s \quad (4.14)$$

定義： $\tau_L = \frac{L}{R T_s}$ ，利用圖 2.6 可推得：

$$I_1 = I_o \left(1 - \frac{1 - D}{2\tau_L}\right) \quad (4.15)$$

$$I_2 = I_o \left(1 + \frac{1 - D}{2\tau_L}\right) \quad (4.16)$$

$$i_{L(rms)} = I_o \left[1 + \frac{1}{12} \left(\frac{1-D}{\tau_L} \right)^2 \right]^{1/2} \quad (4.17)$$

$$i_{Q(rms)} = I_o \left\{ D \left[1 + \frac{1}{12} \left(\frac{1-D}{\tau_L} \right)^2 \right] \right\}^{1/2} \quad (4.18)$$

$$i_{D(rms)} = I_o \left\{ (1-D) \left[1 + \frac{1}{12} \left(\frac{1-D}{\tau_L} \right)^2 \right] \right\}^{1/2} \quad (4.19)$$

$$i_{c(rms)} = I_o \frac{(1-D)}{\sqrt{12\tau_L}} \quad (4.20)$$

$$i_{C_d(rms)} = I_o \left\{ D \left[(1-D) + \frac{1}{12} \left(\frac{1-D}{\tau_L} \right)^2 \right] \right\}^{1/2} \quad (4.21)$$

以上之公式可用以精確計算元件之電流作為選擇元件之依據。

B. 邊界導通模式(BCM)

降壓式轉換器之電感電流的平均值(I_L)等於其負載電流($I_o = V_o/R$)，當負載電流變小使得電感電流在一週期結束時剛好下降至零則稱為 BCM，此時之電感電流稱為臨界電感電流(I_{LB})，相對應之負載電流稱為臨界負載電流電流(I_{OB})，BCM 之工作波形如圖 4.8 所示。在 BCM 下， I_{LB} 滿足：

$$I_{LB} = \frac{1}{2} i_{L,pk} = \frac{1}{2} \frac{(V_d - V_o)}{L} t_{on} = \frac{DT_s}{2L} (V_d - V_o) = I_{OB} \quad (4.22)$$

以下可分為 V_d 為定值與 V_o 為定值之應用來探討：

V_d 為定值

$$V_o = DV_d \quad (4.23)$$

將(4.23)代入(4.22)可得：

$$I_{LB} = \frac{T_s V_d}{2L} D(1-D) \quad (4.24)$$

其最大值發生在 $D = 0.5$ 為：

$$I_{LB,max} = \frac{T_s V_d}{8L} \quad (4.25)$$

因此(4.24)可重新表示為：

$$I_{LB} = 4I_{LB,\max}D(1-D) \quad (4.26)$$

V_o 為定值

$$V_d = V_o / D \quad (4.27)$$

將(4.27)代入(4.22)可得：

$$I_{LB} = \frac{T_s V_o}{2L} (1-D) \quad (4.28)$$

其最大值發生在 $D = 0$ 為：

$$I_{LB,\max} = \frac{T_s V_o}{2L} \quad (4.29)$$

因此(4.28)可重新表示為：

$$I_{LB} = (1-D)I_{LB,\max} \quad (4.30)$$

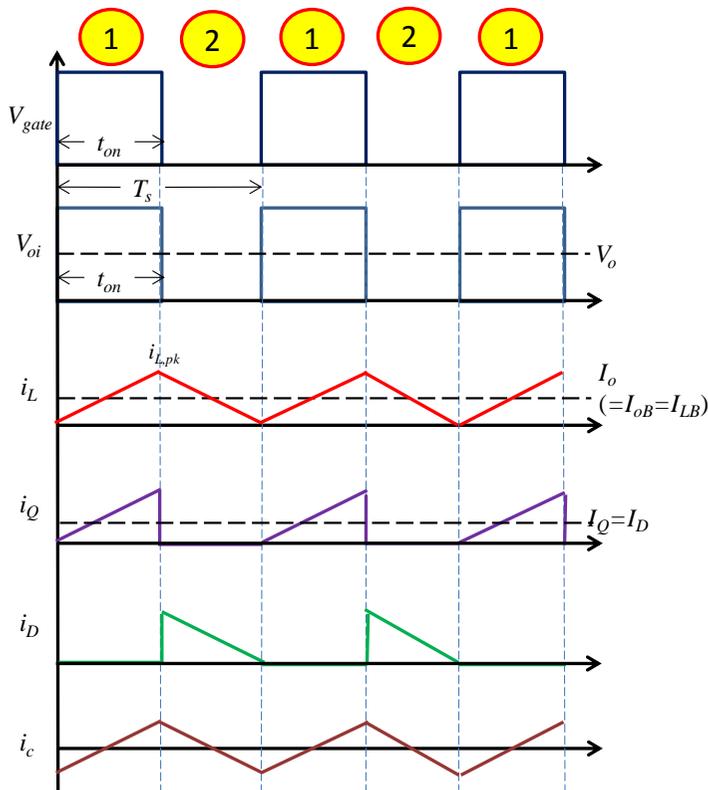


圖 4.7 降壓式轉換器於 BCM 下之工作波形

C. 不連續導通模式(DCM)

當負載電流低於 I_{LB} ，則開關截止時電感電流在一週期結束前便下降到零則進入不連續導通模式，DCM 之工作波形如圖 4.9 所示。在 DCM 下，電感電壓 v_L 仍需滿足伏秒平衡(面積 $A=B$)：

$$(V_d - V_o)DT_s + (-V_o)\Delta T_s = 0 \tag{4.31}$$

重新整理可得：

$$\frac{V_o}{V_d} = \frac{D}{D + \Delta} \tag{4.32}$$

電感電流峰值為：

$$i_{L,pk} = \frac{V_o}{L} \Delta T_s \quad (4.33)$$

$$I_o = i_{L,pk} \frac{D+\Delta}{2} = \frac{V_o T_s}{2L} (D+\Delta) \Delta = \frac{V_d T_s}{2L} D \Delta = 4I_{LB,max} D \Delta \quad (4.34)$$

V_d 為定值

利用(4.25)及(4.34)整理可得：

$$\Delta = \frac{I_o}{4I_{LB,max} D} \quad (4.35)$$

將(4.35)代入(4.32)重新整理可得：

$$\frac{V_o}{V_d} = \frac{D^2}{D^2 + \frac{1}{4}(I_o / I_{LB,max})} \quad (4.36)$$

利用(4.9)、(4.36)及(4.36)所繪之降壓式轉換器特性圖如圖 4.10(a)所示。

V_o 為定值

利用(4.29)及(4.34)整理可得：

$$\Delta = \left(\frac{I_o}{I_{LB,max}} \frac{V_d - V_o}{V_d} \right)^{1/2} \quad (4.37)$$

將(4.37)代入(4.32)重新整理可得：

$$D = \frac{V_o}{V_d} \left(\frac{I_o / I_{LB,max}}{1 - V_o / V_d} \right)^{1/2} \quad (4.38)$$

利用(4.9)、(4.30)及(4.38)所繪之降壓式轉換器特性圖如圖 4.10(b)所示。

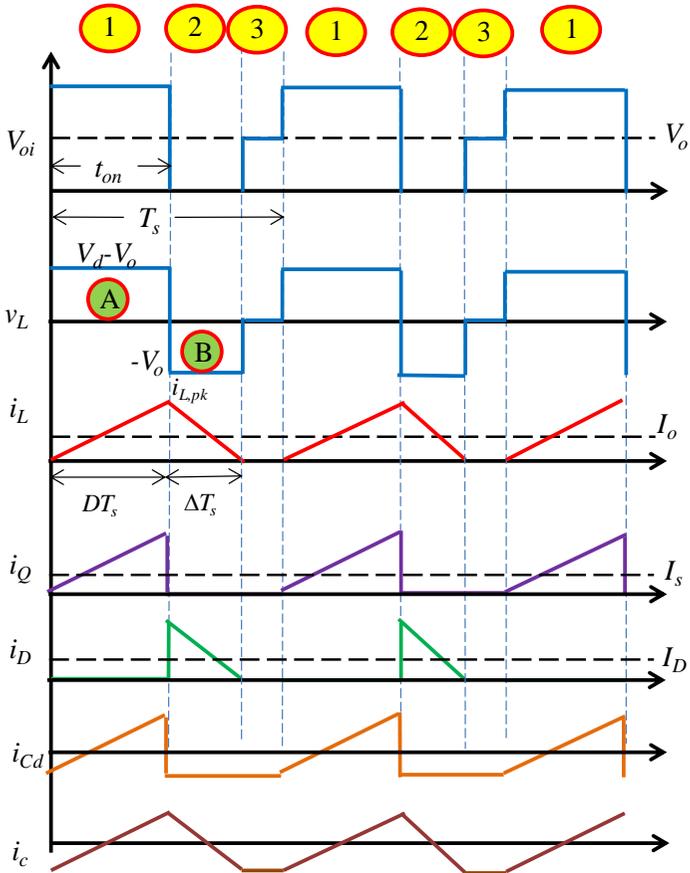
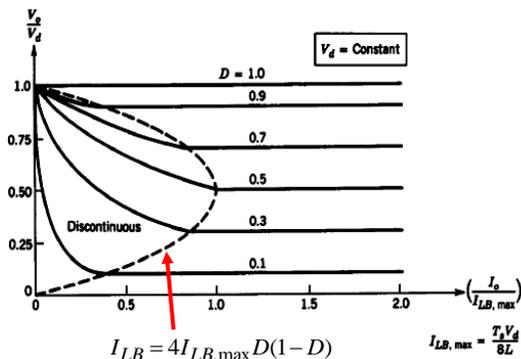


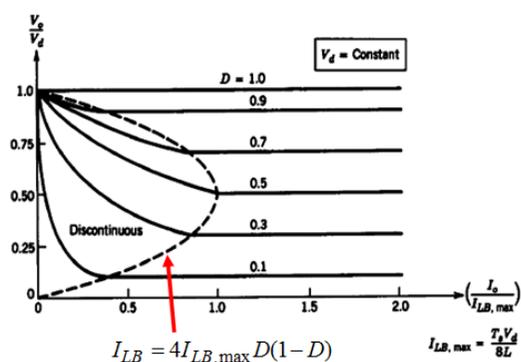
圖 4.8 降壓式轉換器於 DCM 下之工作波形

圖 4.9
降壓式轉換器之
特性圖：

(a) V_d 為定值



(b) V_o 為定值



L 與 C 之設計

(1) 電感設計

方法 1：考慮導通模式

乃以最小電流($I_o = I_{oB}$)當成邊界條件來設計電感，使負載電流高於 I_{oB} 便操作在 CCM，如圖 4.10 所示其滿足：

$$2I_{oB} = i_{L,peak} = \frac{V_d - V_o}{L} T_{on} \tag{4.39}$$

V_d 為定值

考慮輸出電壓變化之責任周期變化，利用(4.39)可得

$$L_{\min} = \frac{T_s V_d}{2I_{oB}} [D(1-D)]_{\max} \quad (4.40)$$

V_o 為定值

考慮輸入電壓變化之責任周期變化，利用(4.39)可得

$$L_{\min} = \frac{T_s V_o}{2I_{oB}} (1-D_{\min}) \quad (4.41)$$

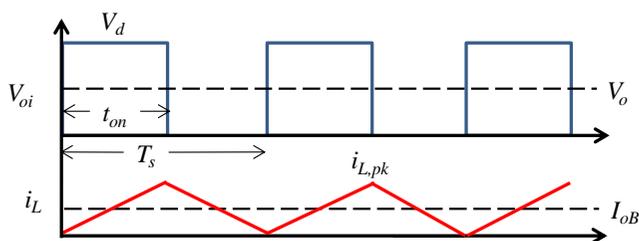


圖 4.10 降壓式轉換器操作於 BCM 之電感電流波形

方法 2：考慮電感電流漣波

訂定一電感電流漣波規格 $I_L/I_{o, rated}$ ，以之求出滿足此規格之電感量。如圖 4.11 所示，操作於 CCM 下之電感電流漣波為：

$$\Delta I_L = \frac{V_d - V_o}{L} t_{on} \quad (4.42)$$

V_d 為定值

考慮輸出電壓變化之責任周期變化，利用(4.42)可得

$$L_{\min} = \frac{T_s V_d}{2\Delta I_L} [D(1-D)]_{\max} \quad (4.43)$$

V_o 為定值

考慮輸入電壓變化之責任周期變化，利用(4.43)可得

$$L_{\min} = \frac{T_s V_o}{2\Delta I_L} (1-D_{\min}) \quad (4.44)$$

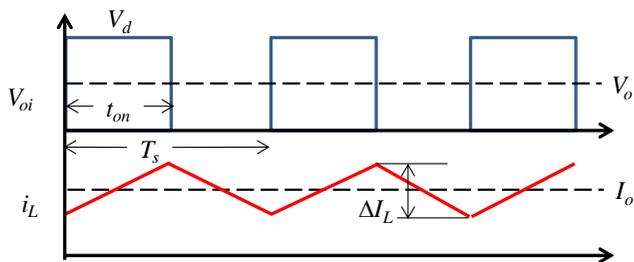


圖 4.11 降壓式轉換器操作於 CCM 之電感電流漣波

以上兩種電感設計方法其實是可等效的，例如 20% 的電感電流漣波規格與 $I_{oB}=10\%I_{o, rated}$ 是相同的。

(2) 電容設計

考慮因素 1：輸出電壓漣波

輸出電容漣波乃由電感電流漣波對電容充放電造成，如圖 4.12 所示，此漣波成份包含電容本身充放電所造成之漣波以及電容電流流經其 ESR(Equivalent Series Resistance，等效串聯電阻)所造成之漣波：

$$\begin{aligned}\Delta V_o &= \Delta I_L R_e + \frac{\Delta Q}{C} \\ &= \Delta V_{o1} + \Delta V_{o2}\end{aligned}\quad (4.45)$$

此電壓漣波成份若由 ESR(R_e)造成之漣波為最主要($\Delta V_o = \Delta V_{o1}$)，則電容之設計必須先以其 ESR 來設計，再由此 ESR 決定電容值：

$$R_e = \frac{\Delta V_o}{\Delta I_L} \quad \text{or} \quad R_e = \frac{\Delta V_o}{2I_{oB}}\quad (4.46)$$

$$C = \frac{\text{(RC product of Capacitor)}}{R_e}\quad (4.47)$$

通常採用電解質電容此電壓漣波成份若由 ESR 造成之漣波為最主要，典型電解質電容(RC product) = $30 \times 10^{-6} \sim 80 \times 10^{-6}$ 。

若採用之電容屬低 ESR 材質，則此電壓漣波成份將由電容本身充放電所造成之漣波為最主要($\Delta V_o = \Delta V_{o2}$)：

$$\begin{aligned} \Delta V_o &= \frac{\Delta Q}{C} = \frac{(\Delta I_L / 2)(T_s / 2)}{C} \\ &= \frac{\Delta I_L T_s}{8C} = \frac{V_o(1-D)T_s^2}{8LC} \end{aligned} \tag{4.48}$$

利用(4.48)可得：

$$C = \frac{V_o(1-D_{\min})T_s^2}{8L\Delta V_o} \tag{4.49}$$

利用(4.49)設計之前必須先訂定電感值。

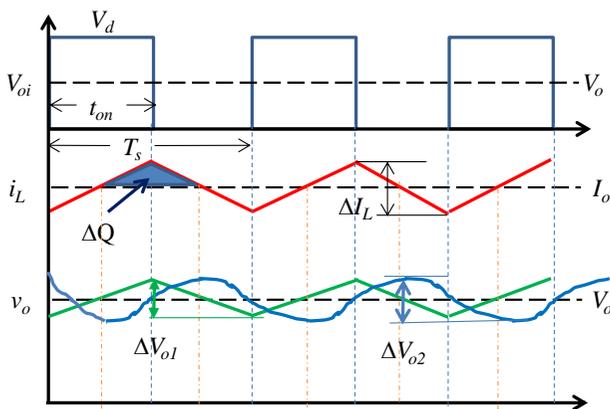


圖 4.12 降壓式轉換器操作於 CCM 之電壓漣波

考慮因素 2：電容儲能能力

當負載突然變化電感電流還來不及補充時，負載電流由電容提供，造成輸出電壓瞬時之下降(亦或減載時為上升)，如圖 4.13 所示，電壓若在 \$\Delta t\$ 時間內由 \$V_{o1}\$ 下降到 \$V_{o2}\$，則滿足以下關係：

$$\Delta P_o = \frac{C(V_{o1}^2 - V_{o2}^2)}{2\Delta t} \tag{4.50}$$

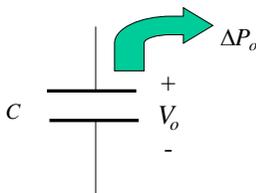
利用(4.50)可求得所需之電容量：

$$C = \frac{2\Delta P_o \Delta t}{V_{o1}^2 - V_{o2}^2} \quad (4.51)$$

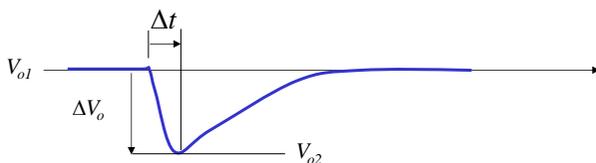
在設計上同常可訂定 ΔV_o 以及 Δt 之規格， ΔP_o 可以 $0.9P_{o,max}$ 來設定。藉由上述電壓漣波規格及功率處理能力所決定之電容值必需取最大值來當成最終之電容值，通常低 ESR 電容者，電容值通常由電容儲能能力決定。

圖 4.13

(a) 負載突然增加電感電流還來不及補充時，負載電流由電容提供



(b) 電容電壓暫態波形



電路模擬

降壓式轉換器電路參數如下：

$V_d=50V$, $V_{tri}=5V_{pp}/40kHz$, $L=365\mu H$, $C=300\mu F$, $R=24*24/10\Omega$,

開迴路控制模擬電路如圖 4.14 所示，模擬結果如圖 4.15 所示。

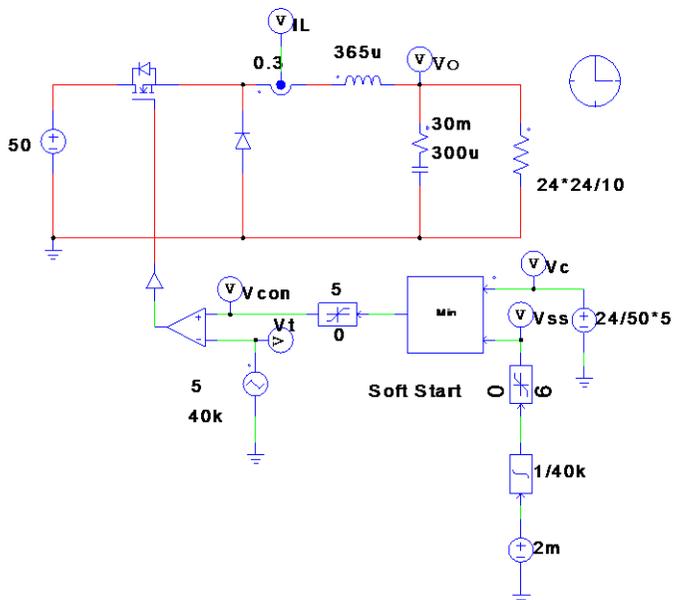


圖 4.14 採用單電壓極性切換之模擬電路

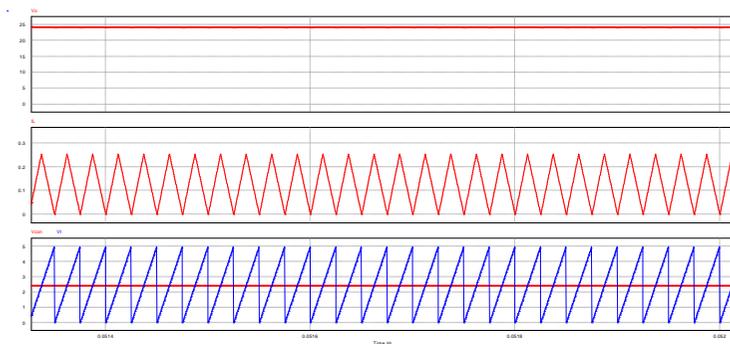


圖 4.15 圖 4.14 採用單電壓極性切換之模擬結果

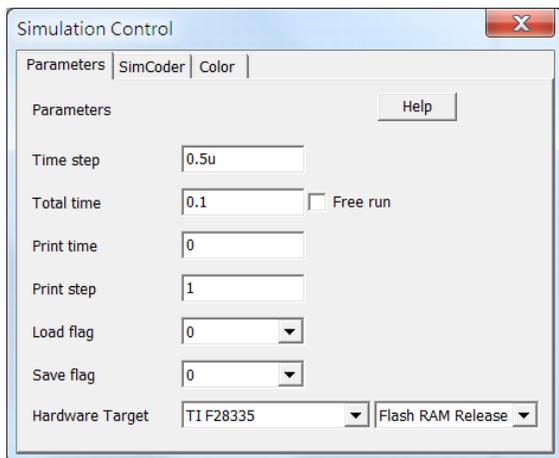
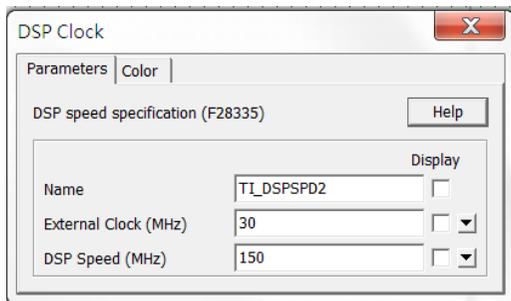


圖 4.16 採用單電壓極性切換之 SimCoder 模擬電路

DSP Clock 設定



DSP Clock 可採用外部設定亦或內建，採用外部設定需自行更改 External Clock 參數(default=30MHz)，採用內建則使用 DSP speed 150MHz。本實驗採用內建值。



28335 硬體設定(Hardware Config)



本實驗 DSP F28335 之硬體設定必須在如下表中以點選打勾方式設定如下

- 使用一組 PWM，PWM1(GPIO 0, GPIO 1)。
- 使用一組 DI (Digital Input，GPIO 49)，可以利用外部電路來啟動 PWM。
- 使用一組串列通訊 SCI (SCI C, GPIO62, 63)用以在電路動作時，將信號透過 RS232-USB 方式傳至電腦，監控系統之工作。

GPIO 0	Digital Input	Digital Output	▼ PWM		
GPIO 1	Digital Input	Digital Output	▼ PWM	Capture	
GPIO 2	Digital Input	Digital Output	PWM		
GPIO 3	Digital Input	Digital Output	PWM	Capture	
GPIO 4	Digital Input	Digital Output	PWM		
GPIO 5	Digital Input	Digital Output	PWM	Capture	
GPIO 6	Digital Input	Digital Output	PWM		
GPIO 7	Digital Input	Digital Output	PWM	Capture	
GPIO 8	Digital Input	Digital Output	PWM		
GPIO 9	Digital Input	Digital Output	PWM	Capture	Serial Port
GPIO 10	Digital Input	Digital Output	PWM		
GPIO 11	Digital	Digital	PWM	Capture	Serial

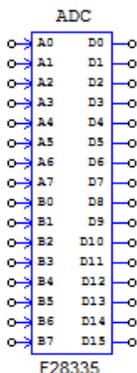
	Input	Output			Port
GPIO 12	Digital Input	Digital Output	Trip-Zone		
GPIO 13	Digital Input	Digital Output	Trip-Zone		
GPIO 14	Digital Input	Digital Output	Trip-Zone	Serial Port	
GPIO 15	Digital Input	Digital Output	Trip-Zone	Serial Port	
GPIO 16	Digital Input	Digital Output	Trip-Zone	SPI	
GPIO 17	Digital Input	Digital Output	Trip-Zone	SPI	
GPIO 18	Digital Input	Digital Output	Serial Port	SPI	
GPIO 19	Digital Input	Digital Output	Serial Port	SPI	
GPIO 20	Digital Input	Digital Output	Encoder		
GPIO 21	Digital Input	Digital Output	Encoder		
GPIO 22	Digital Input	Digital Output	Encoder	Serial Port	
GPIO 23	Digital Input	Digital Output	Encoder	Serial Port	
GPIO 24	Digital Input	Digital Output	PWM	Capture	Encoder
GPIO 25	Digital Input	Digital Output	PWM	Capture	Encoder
GPIO 26	Digital Input	Digital Output	PWM	Capture	Encoder
GPIO 27	Digital Input	Digital Output	PWM	Capture	Encoder
GPIO 28	Digital Input	Digital Output	Serial Port		
GPIO 29	Digital Input	Digital Output	Serial Port		
GPIO 30	Digital Input	Digital Output			
GPIO 31	Digital	Digital			

	Input	Output		
GPIO 32	Digital Input	Digital Output		
GPIO 33	Digital Input	Digital Output		
GPIO 34	Digital Input	Digital Output	PWM	Capture
GPIO 35	Digital Input	Digital Output	Serial Port	
GPIO 36	Digital Input	Digital Output	Serial Port	
GPIO 37	Digital Input	Digital Output	PWM	Capture
GPIO 38	Digital Input	Digital Output		
GPIO 39	Digital Input	Digital Output		
GPIO 40	Digital Input	Digital Output		
GPIO 41	Digital Input	Digital Output		
GPIO 42	Digital Input	Digital Output		
GPIO 43	Digital Input	Digital Output		
GPIO 44	Digital Input	Digital Output		
GPIO 45	Digital Input	Digital Output		
GPIO 46	Digital Input	Digital Output		
GPIO 47	Digital Input	Digital Output		
GPIO 48	Digital Input	Digital Output	PWM	Capture
GPIO 49	▼ Digital Input	Digital Output	PWM	Capture
GPIO 50	Digital Input	Digital Output	Encoder	
GPIO 51	Digital	Digital	Encoder	

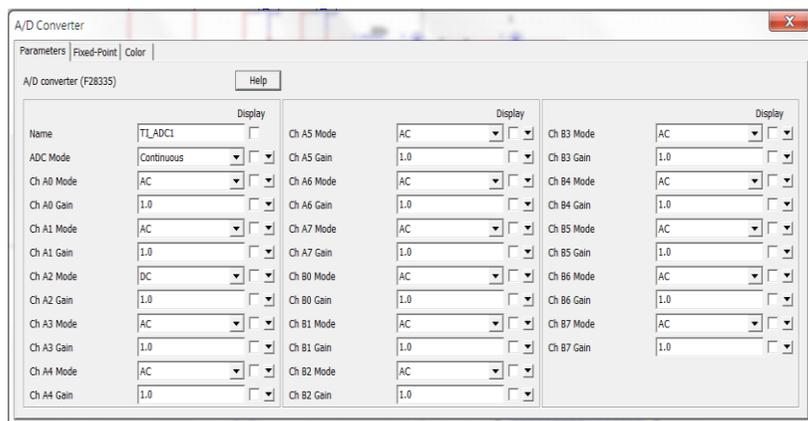
	Input	Output		
GPIO 52	Digital Input	Digital Output	Encoder	
GPIO 53	Digital Input	Digital Output	Encoder	
GPIO 54	Digital Input	Digital Output	SPI	
GPIO 55	Digital Input	Digital Output	SPI	
GPIO 56	Digital Input	Digital Output	SPI	
GPIO 57	Digital Input	Digital Output	SPI	
GPIO 58	Digital Input	Digital Output		
GPIO 59	Digital Input	Digital Output		
GPIO 60	Digital Input	Digital Output		
GPIO 61	Digital Input	Digital Output		
GPIO 62	Digital Input	Digital Output	▼ Serial Port	
GPIO 63	Digital Input	Digital Output	▼ Serial Port	
GPIO 64	Digital Input	Digital Output		
GPIO 65	Digital Input	Digital Output		
GPIO 66	Digital Input	Digital Output		
GPIO 67	Digital Input	Digital Output		
GPIO 68	Digital Input	Digital Output		
GPIO 69	Digital Input	Digital Output		
GPIO 70	Digital Input	Digital Output		
GPIO 71	Digital	Digital		

	Input	Output	
GPIO 72	Digital Input	Digital Output	
GPIO 73	Digital Input	Digital Output	
GPIO 74	Digital Input	Digital Output	
GPIO 75	Digital Input	Digital Output	
GPIO 76	Digital Input	Digital Output	
GPIO 77	Digital Input	Digital Output	
GPIO78	Digital Input	Digital Output	
GPIO 79	Digital Input	Digital Output	
GPIO 80	Digital Input	Digital Output	
GPIO 81	Digital Input	Digital Output	
GPIO 82	Digital Input	Digital Output	
GPIO 83	Digital Input	Digital Output	
GPIO 84	Digital Input	Digital Output	
GPIO 85	Digital Input	Digital Output	
GPIO 86	Digital Input	Digital Output	
GPIO 87	Digital Input	Digital Output	

AD Converter(ADC)設定

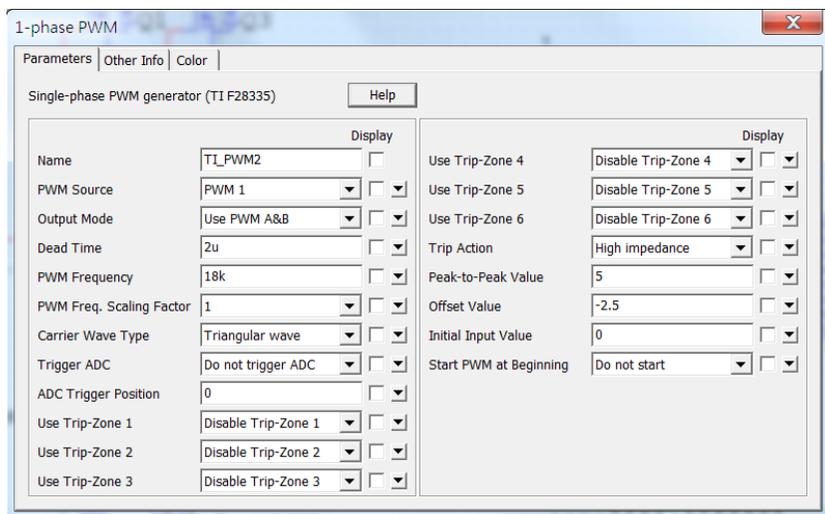
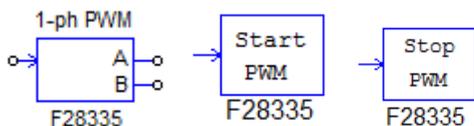


F28335 共有 16 組 ADC，分成 A0~A7 及 B0~B7 等通道，ADC 模組需設定 ADC 取樣模式(ADC Mode)，此處設定連續(Continuous)取樣模式，其次需設定各通道輸入之模式(Mode)以及增益(Gain)，本實驗總計回授 4 信號，分別為負載電流(I_L , A0)、電感電流(I_o , A1)、輸入電壓(V_d , A2)及輸出電壓(V_o , B0)，只有輸入電壓設定為 DC 模式，其餘為 AC 模式，各 ADC 之增益均設為 1。



PWM 設定

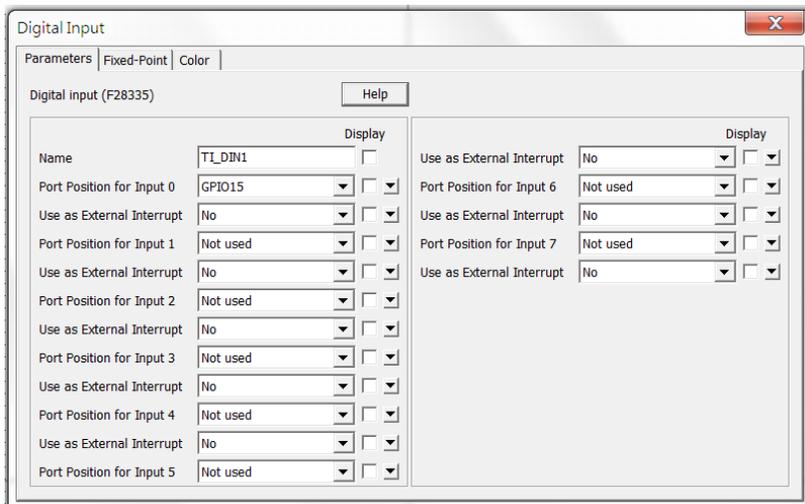
採用單相 PWM，PWM 採用 40kHz 之三角波，PWM 三角波之振幅設定為 $V_p=5V$ ，亦即介於 $0V\sim+5V$ 之三角波。PWM 設為起始狀態不起動，其乃藉由 Start PWM 及 Stop PWM 等二模組來控制其啟動及截止。



PWM

Digital Input(DI)設定

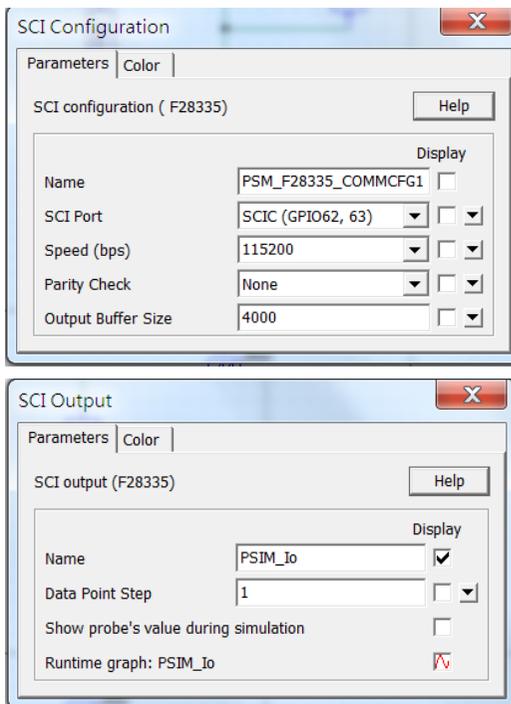
本實驗透過 DI(使用 GPIO 49)用以啟動及關閉 PWM1 及 PWM2。DI 輸入信號乃由 DSP 外部信號所產生。



SCI 設定

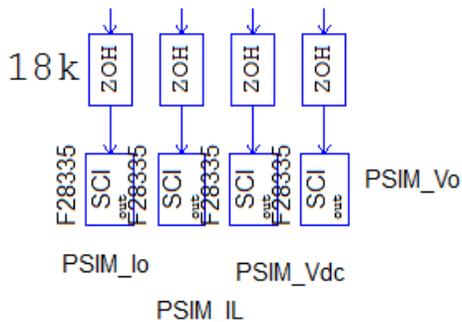
本實驗透過 SCI 將信號傳回電腦，SCI 設定(SCI Config)及 SCI out 二模組之設定如下。SCI 之通訊速度設為 115200(bps)，不使用 parity check，緩衝區大小(buffer size)設定為 4000 點。SCI output 則設定為每一點都傳輸。





注意

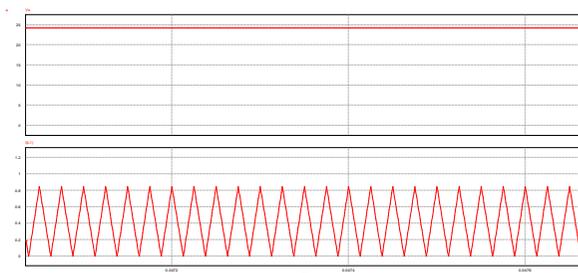
SCI out 模組必須配合 ZOH 模組使用，ZOH 模組用以定義 SCI 之中斷頻率。



SimCoder 電路模擬結果

圖 4.17

圖 4.16 採用
SimCoder 電路之
模擬結果



自動產生 Code

可以利用 PSIM 下之 **Simulate => Generate Code** 將上述轉換器電路之 **SimCoder** 控制電路部分轉換成 **C Code**。PSIM 會在與所模擬電路同一目錄下產生一與模擬檔案名稱相同的子目錄，且將自動產生的 **Code** 及在 **TI Code Composer** 下一 **project** 所需相關的檔案放於此子目錄下。

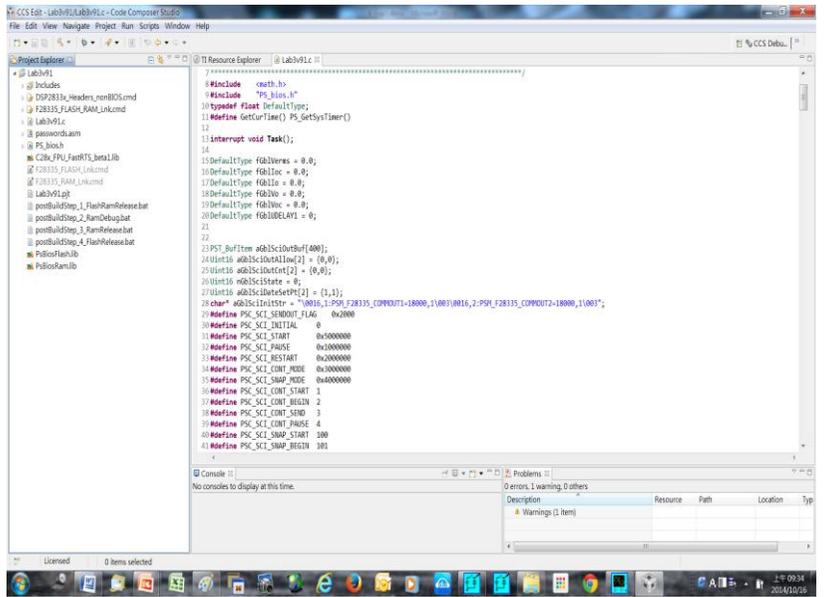
例如：Lab1.psimsch 會產生一子目錄 Lab1

利用 TI Code Composer 進行編譯及燒錄

進入 TI Code Composer Studio

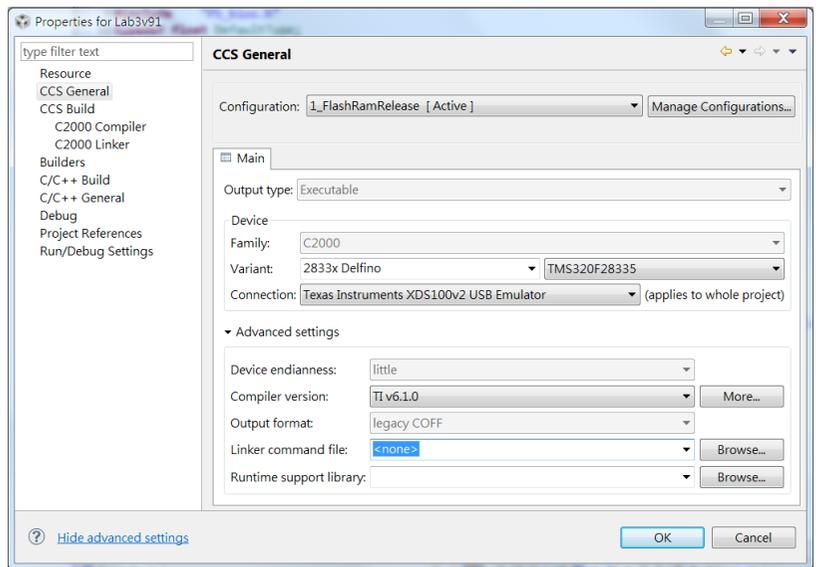
1. 叫出上述次目錄(Lab1)下之 **project: Lab1.pjt**
Project => Import Legacy CCSv3.3 Project => 選擇子目錄下之 **Lab1.pjt => Next => Finish**

便會得到如下畫面，點選 **Lab1.c** 亦可檢視 **SimCoder** 所產生之 **C 檔**：



2. 點選 Lab1[Active, FlashRamRelease] => Build
all 檢查有無錯誤，Warnings 可以忽略。

3. 點選 Lab1[Active, FlashRamRelease] =>滑鼠右鍵
=>作以下設定。



4. RUN => Debug =>將程式燒錄至 DSP IC=>可以拔除 JTAG=>進行實驗

實驗量測

實驗設備與教具配置如圖 4.18，直流電源供應器 PSW 160-7.2 連接到降壓式轉換器教具的輸入端子 J201，其輸出端子 J202 接至電子負載 PEL-2040 並使用定電阻模式，圖 4.19 為輸出電壓與電感電流量測波形。

圖 4.18

實驗設備配置圖

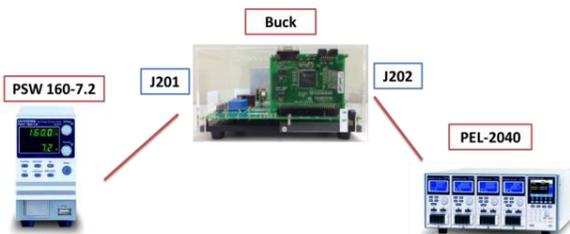
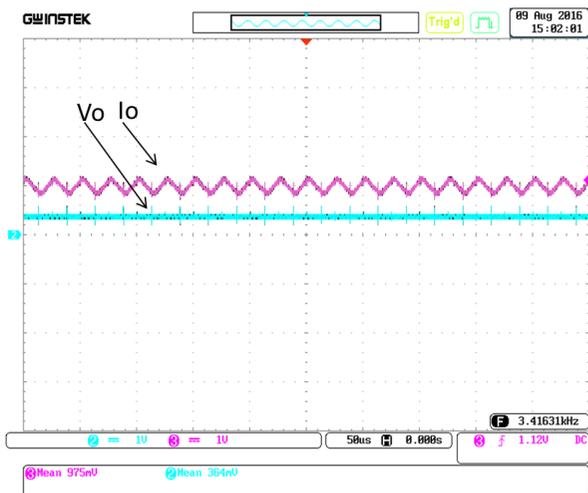


圖 4.19

轉換器輸出電壓
及電感電流之量
測波形



當硬體電路動作後，可以在 PSIM 下打開 DSP Oscilloscope，將 RS232-USB 傳回之信號在電腦上顯示。

打開 DSP Oscilloscope 方法：

Utilities => DSP Oscilloscope =>如圖 4.20，設定 Serial port(可由電腦控制台下之裝置管理員獲得或設定)，Baud rate(需與 SCI Configuration 統一)，Parity check(需與 SCI Configuration 統一) =>設定完成後，再點選 Connect，若連結成功，則會出現如圖 4.21、圖 4.22 之綠燈及傳回之信號畫面。若要存取 DSP Oscilloscope 結果，

操作程序如圖 4.23 所示，點選 Save，選擇存檔目錄位置，接下來便可由 SIMVIEW 來觀察及處理這些信號。

圖 4.20

DSP Oscilloscope 之設定

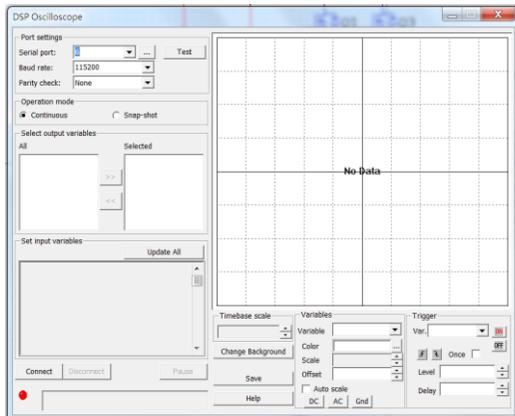


圖 4.21

DSP Oscilloscope 顯示之轉換器輸出電壓及電流之波形

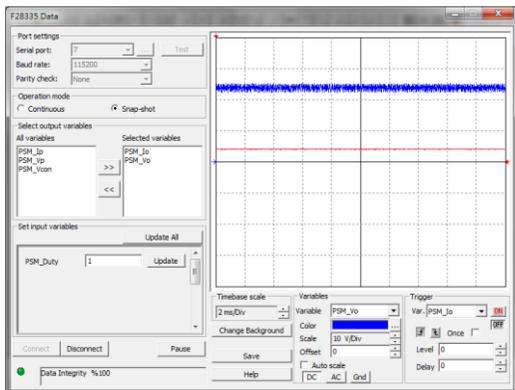


圖 4.22

DSP Oscilloscope 顯示之轉換器輸入電壓與電流之波形

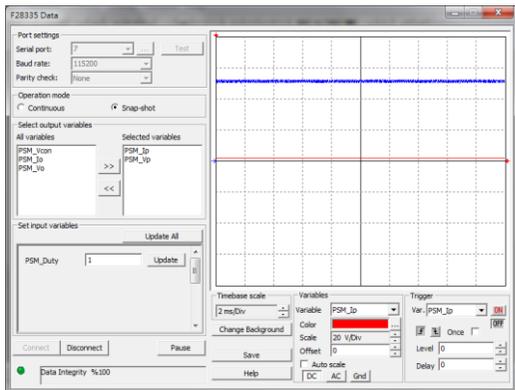
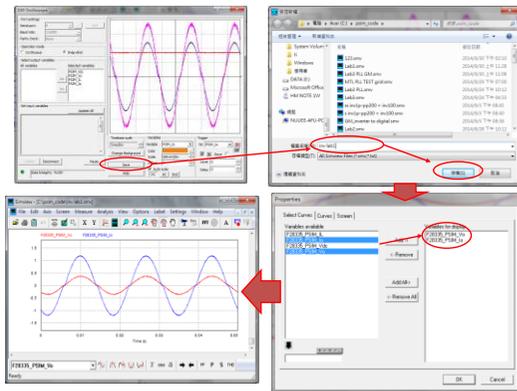


圖 4.23

DSP
Oscilloscope 顯
波形存檔後轉換
至 SIMVIEW 之
方法



實驗 2 電壓模式控制之降壓式轉換器

實驗目的

學習 Buck converter 之小信號模型推導方法、AC sweep 掃波德圖方法，電壓迴路控制器設計方法、硬體規劃及 SimCoder 程式撰寫等。

實驗原理

降壓式轉換器之電壓模式控架構

採用電壓模式控制之降壓式轉換器(Buck Converter)電路架構如圖 5.1 所示，其利用 k_v 對輸出電壓 V_o 分壓取樣(sampling)並經由一低通濾波器(LPF)以得到迴授之輸出電壓 V_{fb} ， V_{fb} 再與參考電壓 V_{ref} 比較，並經由誤差放大器(Error Amplifier, E/A)調整及一限制器(LIM)後得到一控制電壓 V_{con} ， V_{con} 再經由 PWM 比較器與定頻之週期性鋸齒波 V_{st} 比較得到開關之觸發信號 V_{gate} ，藉由脈波 V_{gate} 之責任週期(duty cycle)調整輸出電壓 V_o ，使輸出電壓在負載(R)變化或輸入電壓(V_i)變化下均能維持穩定且精確之電壓。

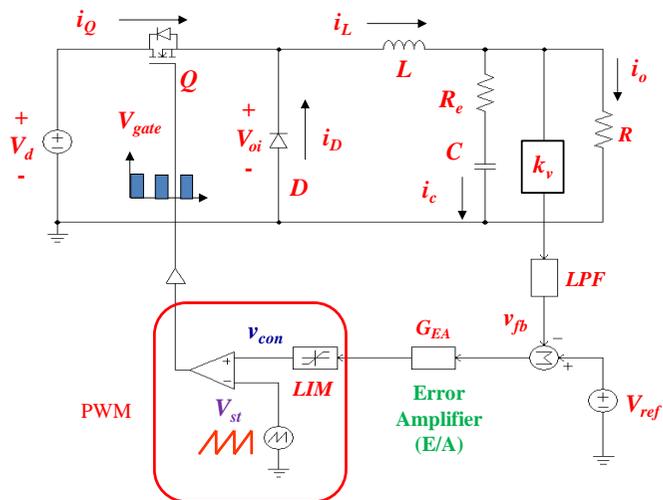


圖 5.1 電壓模式控制之降壓式轉換器

小信號模型

採用 PWM 切換控制之降壓式轉換器可以利用狀態平均法來推導其小信號模型，作為其電壓迴路誤差放大器之設計依據。推導步驟如下：

Step 1: 狀態平均(State Average)

在 CCM 下，降壓式轉換器開關及二極體之狀態平均電路表示如圖 5.2 所示，依據開關導通及截止狀態可以得到以下關係：

$$i_a(t) = i_c(t) \quad 0 < t < dT_s$$

$$= 0 \quad dT_s < t < T_s$$
(5.1)

$$v_{cp}(t) = v_{ap}(t) \quad 0 < t < dT_s$$

$$= 0 \quad dT_s < t < T_s$$
(5.2)

將輸入電流(i_a)及輸出電壓(v_{cp})利用狀態平均可得：

$$i_a = di_c$$
(5.3)

$$v_{cp} = dv_{ap}$$
(5.4)

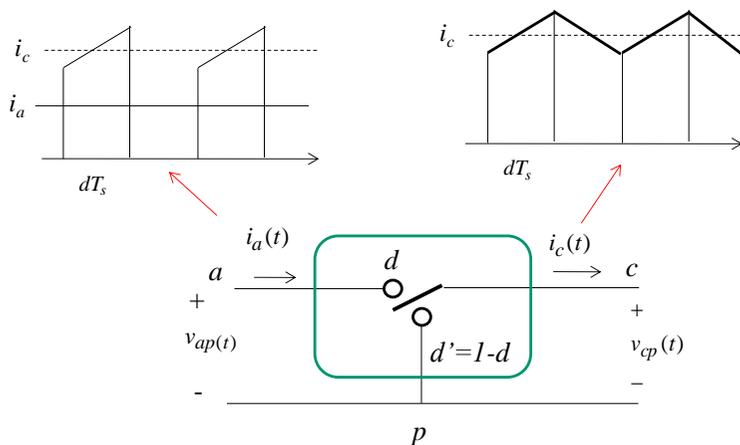


圖 5.2(a)

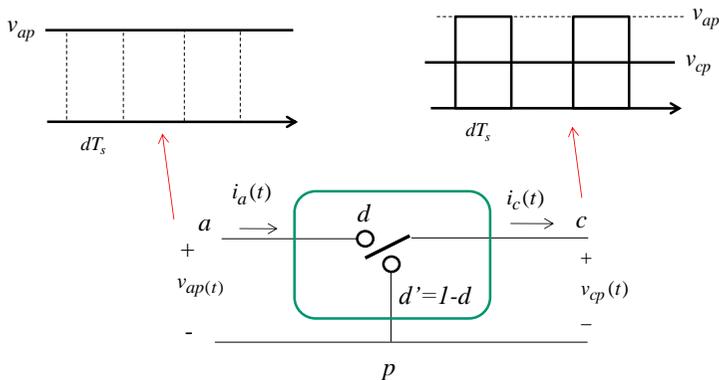


圖 5.2(b)

圖 5.2 降壓式轉換器開關及二極體之狀態平均電路表示

Step 2: 在工作點上加擾動(perturbation)以求得此穩態工作點之小信號模型

$$i_a = I_a + \tilde{i}_a = di_c = (D + \tilde{d})(I_c + \tilde{i}_c) = DI_c + D\tilde{i}_c + \tilde{d}I_c + \tilde{d}\tilde{i} \quad (5.5)$$

$$v_{cp} = V_{cp} + \tilde{v}_{cp} = dv_{ap} = (D + \tilde{d})(V_{ap} + \tilde{v}_{ap}) = DV_{ap} + D\tilde{v}_{ap} + \tilde{d}V_{ap} + \tilde{d}\tilde{v}_{ap} \quad (5.6)$$

其中：

I, V, D : 直流穩態工作點

$\tilde{i}, \tilde{v}, \tilde{d}$: 交流小信號擾動量

利用(2.5)及(2.6)之直流項可以得到電路的穩態工作點。利用(5.5)及(5.6)之交流項並忽略其中之二小信號相乘項，則可以得到電路的小信號方程式如下：

$$\tilde{i}_a = D\tilde{i}_c + I_c\tilde{d} \tag{5.7}$$

$$\tilde{v}_{ap} = \frac{\tilde{v}_{cp}}{D} - \frac{V_{ap}}{D}\tilde{d} \tag{5.8}$$

利用(5.7)及(5.8)可繪出開關及二極體電路之小信號模型電路如圖 5.3 所示。

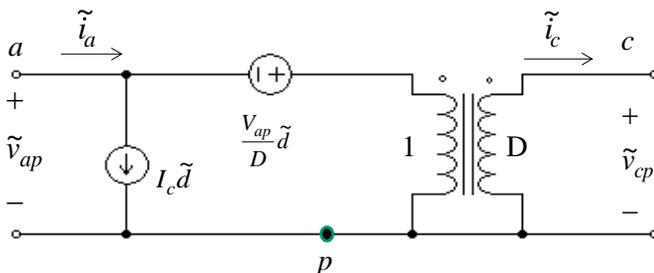


圖 5.3 開關及二極體電路之小信號模型

Step 3: 利用開關及二極體等效電路與輸出濾波器結合求得降壓式轉換器在 CCM 下之小信號模型

利用圖 5.3 開關及二極體之等效電路與輸出 LC 濾波器結合可求得降壓式轉換器在 CCM 下之小信號模型如圖 5.4 所示。

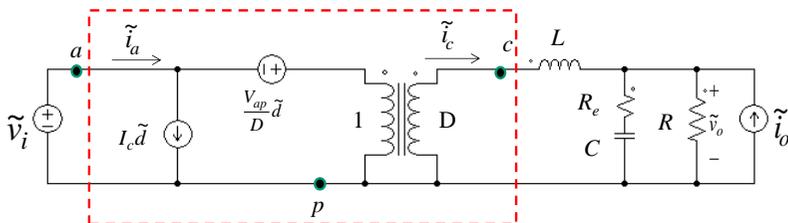


圖 5.4 降壓式轉換器在 CCM 下之小信號模型

利用圖 5.4 可求得以下之轉移函數：

- 控制到輸出轉移函數 (Control to output transfer function, used in control loop design)

$$G_{vd}(s) = \frac{\tilde{v}_o}{\tilde{d}} \Big|_{\tilde{v}_i = \tilde{i}_o = 0} \quad (5.9)$$

- 輸出阻抗 (Output impedance, used in transient response design)

$$Z_o(s) = \frac{\tilde{v}_o}{\tilde{i}_o} \Big|_{\tilde{v}_i = \tilde{d} = 0} \quad (5.10)$$

- 輸入到輸出轉移函數 (Line to output, used in input voltage ripple rejection design)

$$G_{vd}(s) = \frac{\tilde{v}_o}{\tilde{v}_i} \Big|_{\tilde{d} = \tilde{i}_o = 0} \quad (5.11)$$

圖 5.4 中降壓式轉換器在 CCM 下之小信號模型中輸出電壓 \tilde{v}_o 受到三擾動項之影響，包括輸入電壓 \tilde{v}_i 、負載電流 \tilde{i}_o 及責任周期 \tilde{d} 等。欲得控制到輸出之轉移函數可令 $\tilde{v}_i = \tilde{i}_o = 0$ ，並由圖 5.4 求得圖 5.5 之等效電路，由圖 5.5 可得控制到輸出之轉移函數：

$$G_{vd}(s) = \frac{\tilde{v}_o}{\tilde{d}} = V_i \frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{Q\omega_o} + \frac{s^2}{\omega_o^2}} \quad (5.12)$$

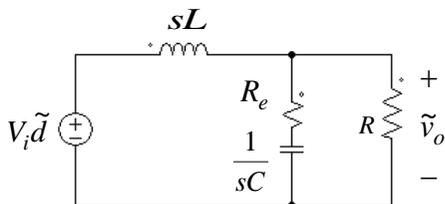
其中

$$\omega_z = \frac{1}{R_e C}, \quad \omega_o = \frac{1}{\sqrt{LC}}, \quad Q = \frac{1}{\omega_o} \frac{1}{\frac{L}{R} + R_e C} \approx \frac{R}{\omega_o L} \quad (5.13)$$

其中之極點 ω_o 由輸出 LC 濾波器決定，零點 ω_z 則由輸出電容及其 ESR 決定。

圖 5.5

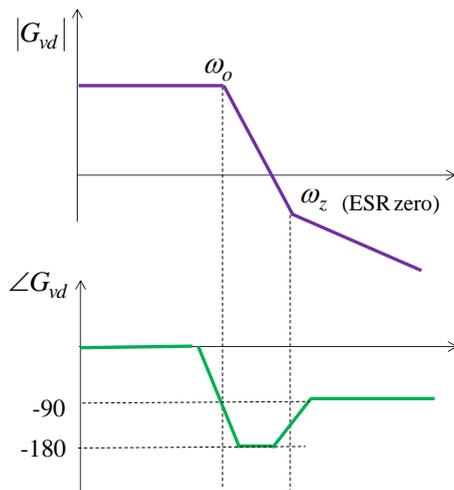
等效電路用以求得控制到輸出之轉移函數



利用(5.12)所繪控制到輸出轉移函數之波德圖如圖 5.6 所示。

圖 5.6

控制到輸出轉移函數之波德圖



欲得輸出阻抗之轉移函數可令 $\tilde{v}_i = \tilde{d} = 0$ ，並由圖 5.4 求得圖 5.7 之等效電路，由圖 5.7 可得輸出阻抗之轉移函數：

$$Z_o(s) = sL \frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{Q\omega_o} + \frac{s^2}{\omega_o^2}} \quad (5.14)$$

$$\omega_z = \frac{1}{R_e C}, \omega_o = \frac{1}{\sqrt{LC}}, Q = \frac{1}{\omega_o} \frac{1}{\frac{L}{R} + R_e C} \approx \frac{R}{\omega_o L} \quad (5.15)$$

利用(5.14)所繪之輸出阻抗轉移函數之波德圖如圖 5.8 所示。

圖 5.7

等效電路用以求得輸出阻抗之轉移函數

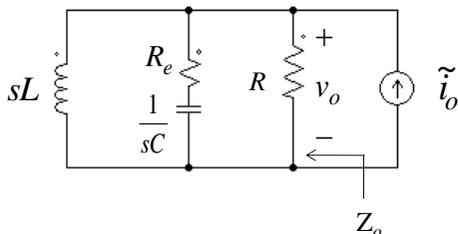
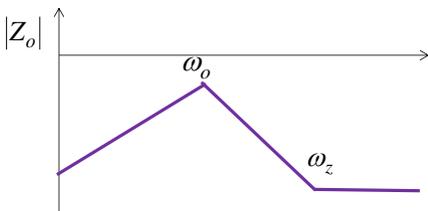


圖 5.8

輸出阻抗之波德圖



欲得輸入到輸出之轉移函數可令 $\tilde{i}_o = \tilde{d} = 0$ ，並由圖 5.4 求得圖 5.9 之等效電路，由圖 5.9 可得輸入到輸出之轉移函數：

$$G_{vv}(s) = \frac{\tilde{v}_o}{\tilde{v}_i} = D \frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{Q\omega_o} + \frac{s^2}{\omega_o^2}} \quad (5.16)$$

$$\omega_z = \frac{1}{R_e C}, \quad \omega_o = \frac{1}{\sqrt{LC}}, \quad Q = \frac{1}{\omega_o} \frac{1}{\frac{L}{R} + R_e C} \approx \frac{R}{\omega_o L} \quad (5.17)$$

利用(5.16)所繪之輸入到輸出轉移函數之波德圖如圖 5.10 所示。

圖 5.9

等效電路用以求得輸入到輸出之轉移函數

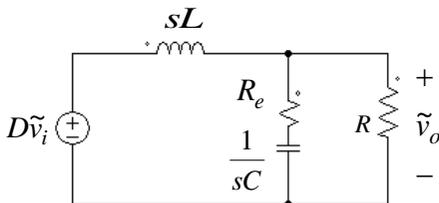
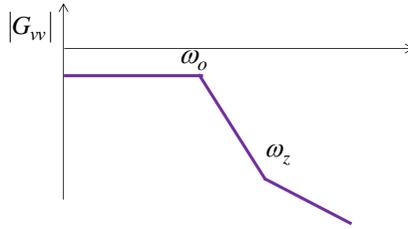


圖 5.10

輸入到輸出轉移
函數之波德圖



迴路增益與電壓誤差放大器

採用電壓模式控制之降壓式轉換器控制方塊圖如圖 5.11 所示，其迴授輸出電壓(sampling 取樣方塊)，再利用迴授電壓(V_{FB})與參考電壓(V_{ref})比較，經由誤差放大器調整得到 PWM 之控制電壓(V_{con})，控制電壓再由 PWM 與鋸齒波(V_t)比較，得到開關之觸發信號。其中 G_s 為取樣電路之增益， G_s 為誤差放大器之轉移函數， G_s 為 PWM 之增益， G_{vd} 即前述控制到輸出之轉移函數，利用圖 5.11 可得圖 5.12 之控制迴路方塊圖，利用前述之轉移函數，可以繪出細部之控制迴路方塊圖如圖 5.13 所示。

圖 5.11

採用電壓模式控制之降壓式轉換器控制方塊圖

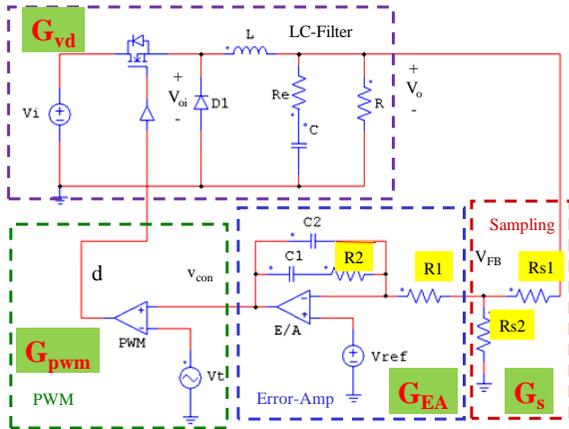


圖 5.12

電壓模式控制之
控制迴路方塊圖

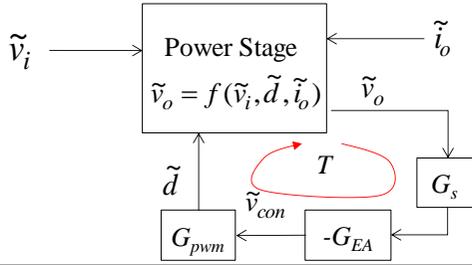
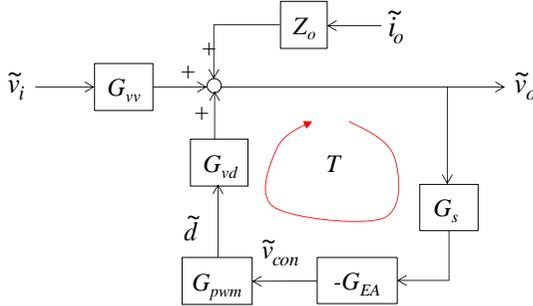


圖 5.13

電壓模式控制之
控制迴路細部分
塊圖



迴路增益(Loop Gain)

電壓閉迴路增益如圖 5.13 所示為：

$$T = G_s G_{EA} G_{pwm} G_{vd} \tag{5.18}$$

傳統電壓誤差放大器(GEA)之設計，乃根據此迴路增益 T 來設計， T 之波德圖如圖 5.14 所示，其經過零 dB 點之頻率稱為零交越頻率(f_{co} , zero crossover frequency)，相當於電壓控制迴路之頻寬(-3dB 與 0dB 相近)。零交越點之相位與 -180° 之差值稱為相位邊限(phase margin, PM)，相位經過 -180° 之頻率下， T 之增益與 0db 之差距稱為增益邊限(gain margin, GM)。

迴路增益需求

- 低頻增益越大，電壓調整率(voltage regulation) 越佳
- 頻寬越寬電壓響應越快
- PM>45
- GM>10~20dB

G_{EA} 乃使 T 能夠滿足上述需求來加以設計。

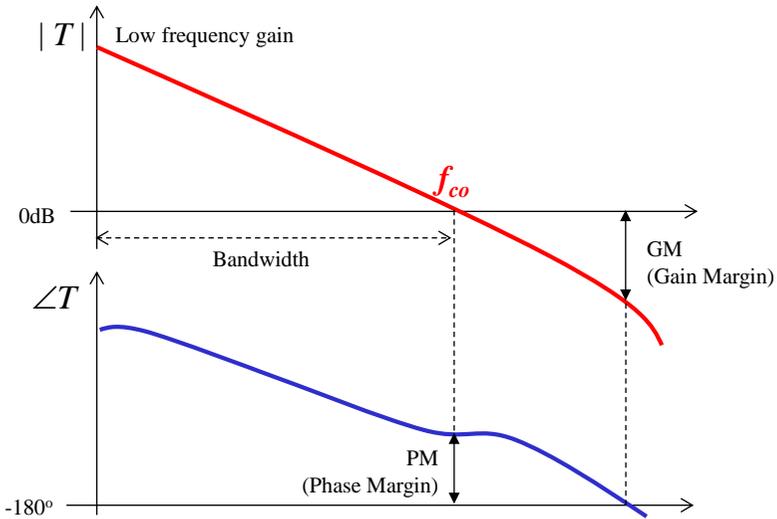


圖 5.14 迴路增益波德圖

實驗系統設計

實驗系統控制方塊與控制器設計

實驗電路電壓模式控制方塊圖如圖 5.15 所示，其中 G_{pwm} 代表轉換器控制電壓到轉換器輸出之增益，LPF1 為低通濾波器用以代表電壓感測電路所加之低通濾波器(頻寬為 16kHz, $p_1=2\pi*16k \text{ rad/s}$)，LPF2 為低通濾波器用以代表電壓感測元件本身之頻寬(頻寬為 10kHz, $p_2=2\pi*10k \text{ rad/s}$)，以上二低通濾波器為製作電路需要，無法避免，為了維持迴路之穩定度，需要利用二零點來補償，FF 為所設計之前向控制器，其包含一零點(z_1)及一極點(p_3)，GV 為比例積分(PI)控制器，除了積分外，尚包含了另一零點(z_2)。切換頻率為 40kHz，控制迴路設計之 Matlab 程式如下所示，根據圖 5.15 之 H1 及 FB 特性設定零交越頻率為 $f_{co} = 700$ ，FF 及 Gv 之參數設定為 $z_1 = 200 * 2\pi$ ， $z_2 = 300$ ， $p_3 = 10e3 * 2\pi$ ，由之可得 Km 之值。根據這些設計所繪出之波德圖如圖 5.16 所示，其確實滿足上述設計， f_{co} 確實為 700Hz， $PM=80^\circ$ 。

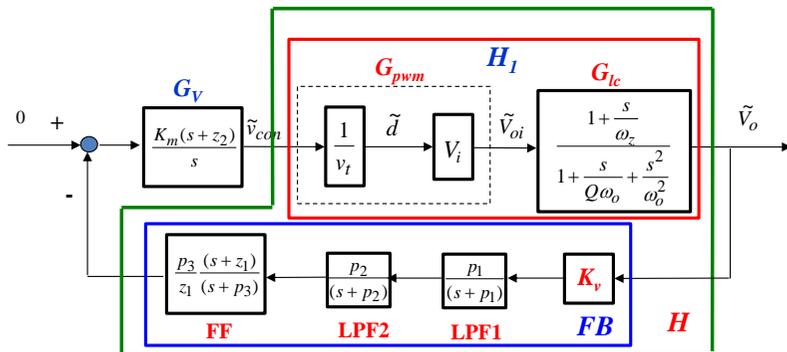


圖 5.15 實際電路之電壓模式控制方塊圖

程式：Buck2.m

% Buck with Type 3 Voltage Mode Control

clf;

clc;

PI = 3.1416;

Vd=50;

Vo=24;

Po=100;

Io=Po/Vo;

Iomin = Io * 0.1;

fs=40e3;

ws = 2 * PI * fs;

Ts=1/fs;

Vt=5;

% PWM

D = Vo/Vd;

Vcon = D * Vt;

% LC

R = Vo/Io;

L = Ts * Vo * (1-D)/(2*Iomin);

L = 365e-6;

Re = 0.13/3;

C = 300e-6;

% Small signal model

Gpwm = Vd/Vt;

wz = 1/(Re * C);

fz = wz/(2*PI)

wo = 1/sqrt(L*C);

fo = wo/(2*PI)

```

Q = R/(wo*L);
numLC = [1/wz 1];
denLC = [1/wo^2 1/(Q*wo) 1];
numH = Gpwm * numLC;
denH = denLC;
H1=tf(numH, denH);
% Controller Design
fco =700
wco = 2 * PI * fco;
Hr = freqresp(H, wco);
GainH = abs(Hr);
Hdb = 20 * log10(GainH);
GainGv = 1/GainH;
Kv = 1/40;
p1 = 16e3 * 2 * PI
p2 = 10e3 * 2 * PI
numFB1 = Kv * p1;
denFB1 = [1 p1];
FB1=tf(numFB1, denFB1);
numFB2 = p2;
denFB2 = [1 p2];
FB2=tf(numFB2, denFB2);
FB = series(FB1, FB2);
H = series(H1, FB);
z1= 200* 2 * PI
z2 =300
p3 = 10e3 * 2 * PI
tu = 1/z2
Am = p3/z1
% Type 3 Gv(s)= Am*(s+z1)/(s+p3) * Km* (s+z2)/s
% Type 3 Gv(s)= FF* PI
% FF = p3/z1*(s+z1)/(s+p3) = Am * (s+z1)/(s+p3)
% PI = Km * (s+z2)/s
numFF = Am * [1 z1];
denFF = [1 p3];
FF = tf(numFF, denFF);
numGv1 = [1 z2];
denGv = [1 0];
Gv1 = tf(numGv1, denGv);
Gv2 = series(Gv1, FF);
Gv2r = freqresp(Gv2, wco);
GainGv2r = abs(Gv2r);
Km = GainGv/GainGv2r

```

```
Gv = Km * Gv2;
GvH = series(Gv, H);
GvHr = freqresp(GvH, wco);
phaseGvH = angle(GvHr) * 180/PI;
wmin = 1 * 2 * PI;
wmax = 100e3 * 2 * PI;
bode(H, Gv, GvH, {wmin, wmax});
grid;
```

=>

```
fz = 1.2243e+004
fo = 480.9634
fco = 700
p1 = 1.0053e+005
p2 = 62832
z1 = 1.2566e+003
z2 = 300
p3 = 62832
tu = 0.0033
Am = 50.0000
Km = 1.2688
phaseGvH = -103.0929
```

Matlab 之設計結果整理如下所示：

電路參數	$V_d=50,$ $V_o=24,$ $f_s=40\text{kHz},$ $V_r=5,$ $L = 365\mu\text{H},$ $R_e = 0.13/3\Omega,$ $C = 300\mu\text{F};$
------	--

迴授參數	$K_v=1/40,$
------	-------------

$$\text{LPF1} = \frac{(s + 2\pi \times 16k)}{s}, \quad \text{LPF2} = \frac{(s + 2\pi \times 10k)}{s},$$

$$\text{FF} = \frac{62832}{1256} \frac{(s + 1256)}{(s + 62832)} = 50 \frac{(s + 1256)}{(s + 62832)}$$

控制器參數

$$G_v = \frac{1.2688(0.0033s + 1)}{0.0033s}$$

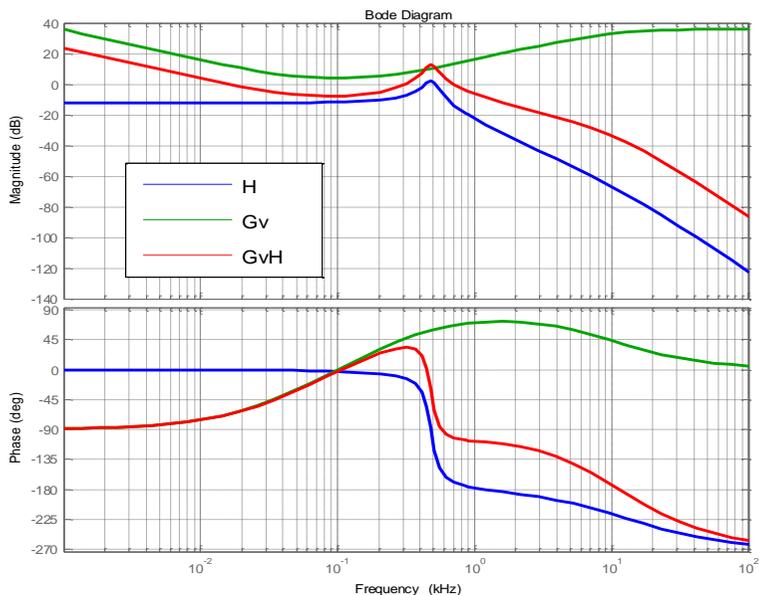


圖 5.16 電壓控制迴路波德圖

電路模擬

根據上述設計結果所建置之 PSIM 模擬電路如圖 5.17 所示，其包含一緩啟動電路，利用電壓命令直線上升來進行(VSS)，其與原電壓命令經由一最小值選擇方塊來選擇最終之電壓命令(VC)，50W 變化至 100W 之模擬結果如圖 5.18 所示。緩啟動電路需要根據啟動時間、啟動電流等要求來調整，可以更改 1m 之值來調整。

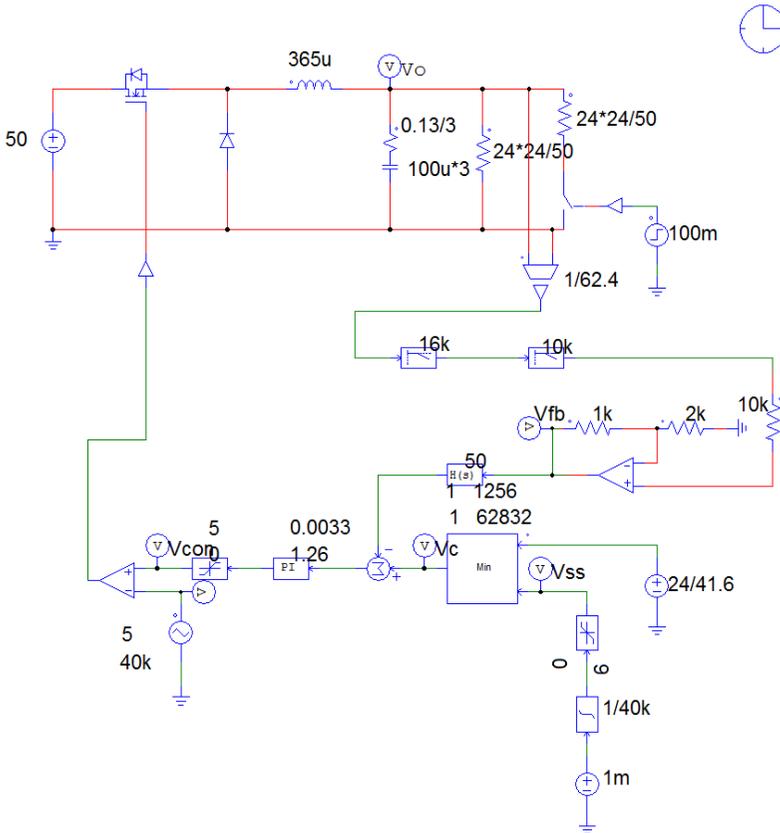


圖 5.17 Buck converter 電壓模式控制之類比 PSIM 模擬電路
(buck2.psimsch)

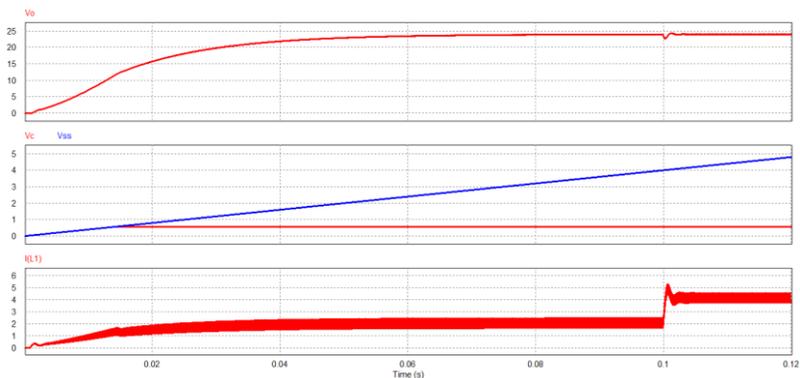


圖 5.18 (a)

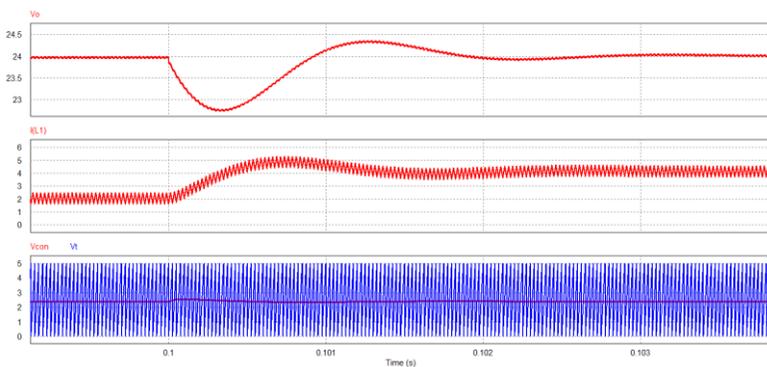


圖 5.18 (b)

圖 5.18 Buck converter 電壓模式控制之類比 PSIM 模擬

為了驗證控制迴路設計之正確性，圖 5.19 進行電壓迴路之 AC sweep，其掃描結果如圖 5.20 所示，其結果與圖 5.16 Matlab 之波德圖吻合。

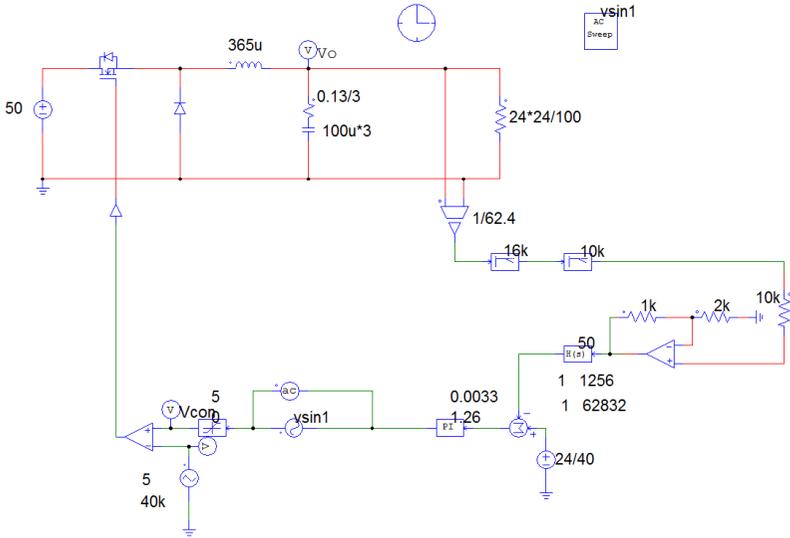


圖 5.19 Buck converter 之電壓迴路 AC sweep (buck2_AC.psimsch)

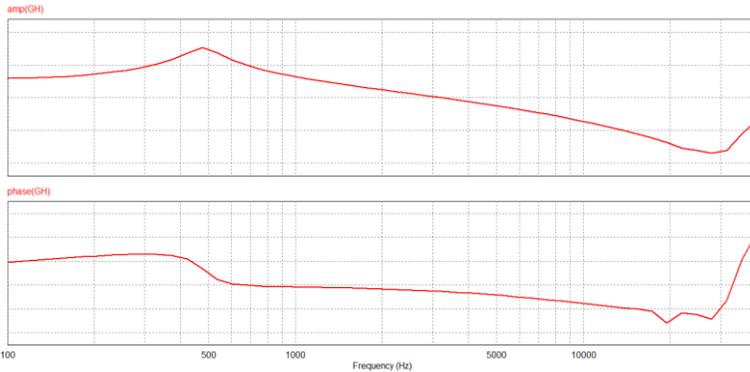


圖 5.19 Buck converter 之電壓迴路 AC sweep 結果

圖 5.21 所示為 Buck convert 除定電壓(constant voltage, CV)控制外，另外再加入定電流(constant current, CC)控制之 PSIM 模擬程式，定電流之準位設定為 5A。負載電阻由 6Ω(24V 下為 4A)變化到 3Ω(24V 下為 8A)之模擬結果如圖 5.22 所示，當負載電流等於 4A 時 CV 動作，電壓維持在 24V，當負載切換至 3Ω時，由於超過 5A，CC 動作使輸出電流限制在 5A，因此輸出電壓降低至 15V(=3Ω x 5A)。

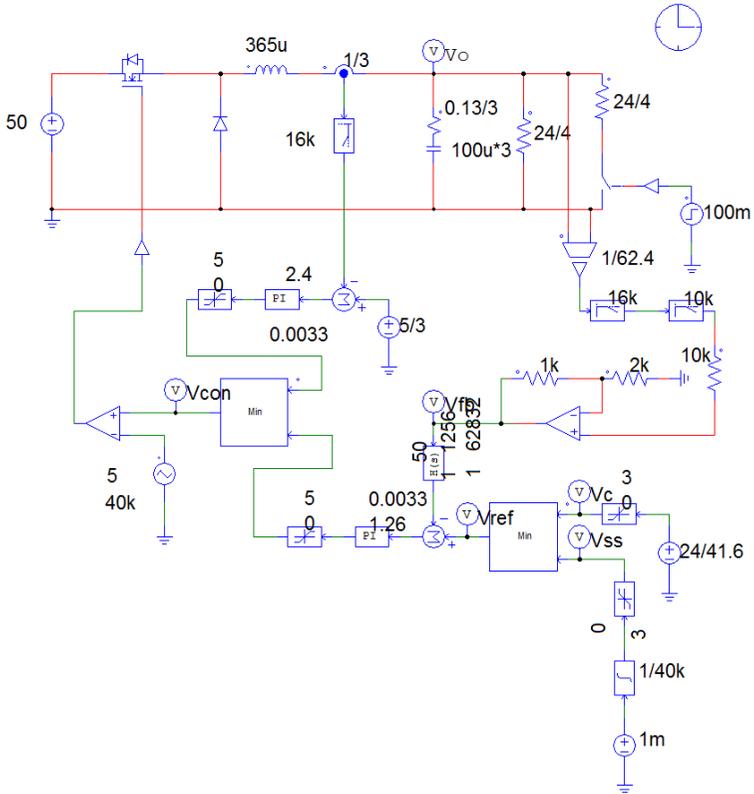


圖 5.21 Buck converter 包含 CV 及 CC 控制之 PSIM 模擬程式
(buck_2_I_Limit.psim.sch)

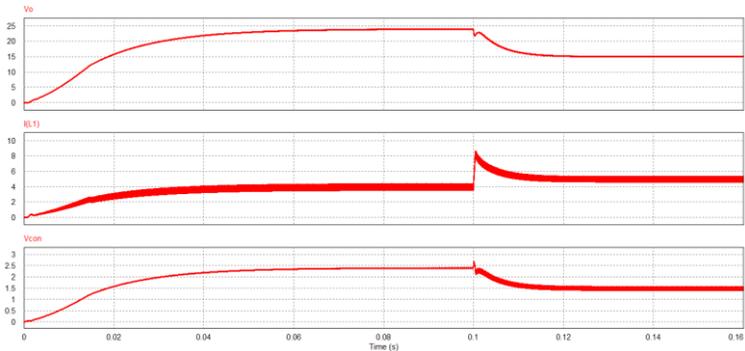


圖 5.22 CV 及 CC 之模擬結果

SimCoder 程式規劃

圖 5.23 所示為 Buck converter 之電壓模式控制 SimCoder 程式，其乃將前述之類比模擬電路利用 TI F2833X Target 及 PSIM 之一些可以轉成 Code 之方塊所建立。其中之 G_V (PI 控制器)及 $FF(\frac{p_3(s+z_1)}{z_1(s+p_3)})$ 可

以利用 PSIM>Utility>s2z Converter 來協助轉成數位控制方塊。PI 之 s 到 z 轉換如圖 5.24 所示，使用技巧上有三點需要注意：(1)建議使用 Back Euler 之數位轉換方式，(2)積分(即累加)需要加一限制器，此限制器必須要有正有負，而且要能達到 PI 後之限制值。例如本範例 PI 之限制值為 0~5，因此積分器之限制值設定為-5~+5。(3)積分(累加)信號必須乘上一啟動(RUN)信號，其在 PWM 啟動後開始動作，但在 PWM 停止後則會被重置(reset)為零。

FF 之 s 到 z 轉換則如圖 2.25 所示，其使用 General (1st-Order)之 s-Domain Function，並使用 Bilinear(Tunstin)之轉換方法，s-Domain 之參數設定則與類比設定相同，取樣頻率等於切換頻率 $f_s=40\text{kHz}$ 。

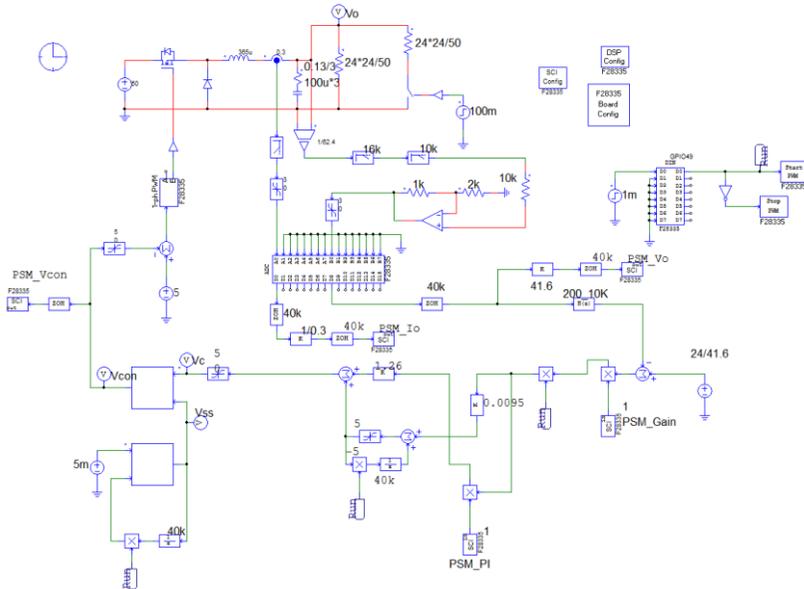


圖 5.23 Buck converter 之電壓模式控制 SimCoder 程式 (Lab2.psim.sch)

s2z Converter

s-Domain Function

- Integrator
- Proportional-Integral
- Modified PI
- Single-Pole / 1st-Order Low-Pass Filter
- 2nd-Order Low-Pass Filter
- 2nd-Order High-Pass Filter
- 2nd-Order Band-Pass Filter
- 2nd-Order Band-Stop Filter
- General (1st-Order)
- General (2nd-Order)
- General (3rd-Order)

Sampling Frequency f_s (period $T_s = 1/f_s$)

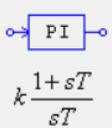
40k (Hz)

Conversion Method

- Bilinear (Tustin)
- Backward Euler

s-Domain Function

Block Diagram



$$k \frac{1 + sT}{sT}$$

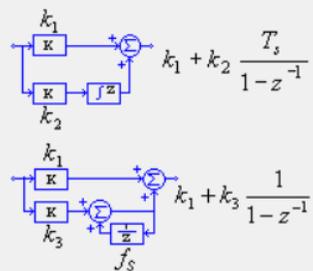
Parameters

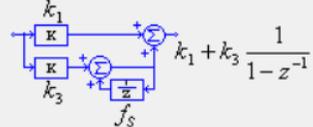
$k =$ 1.26

$T =$ 0.0033

z-Domain Function

Block Diagram



$$k_1 + k_2 \frac{T_s}{1 - z^{-1}}$$


$$k_1 + k_3 \frac{1}{1 - z^{-1}}$$

Parameters

k1 = 1.26

k2 = 381.81818

k3 = 0.0095454545

Convert

Close

圖 5.24 (a)

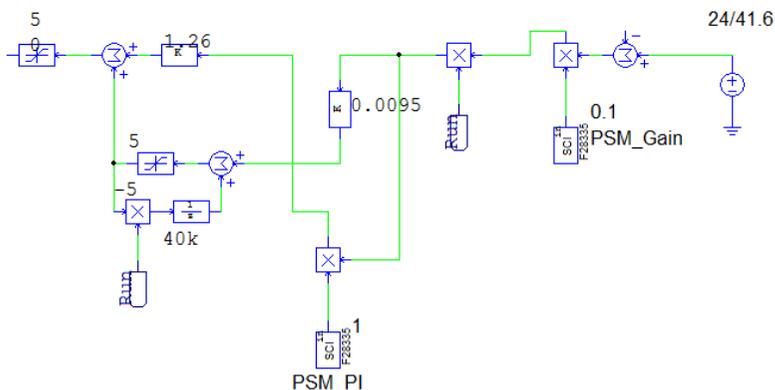


圖 5.24 (b)

圖 5.24 PI 之 s 到 z 轉換：(a) s2z Converter 設定，(b) SimCoder 程式

圖 5.25 FF 之 s 到 z 轉換

圖 5.23 中具備 Start PWM 與 Stop PWM 之設定，由 GPIO 49 之 DI 所設定，此 DI 輸入連接至硬體之啟動開關，開關啟動後才啟動 PWM，模擬時假設 1ms 時開關被啟動。其次與類比不同的是，緩啟動乃由控制電壓之限制來執行，兩種方法均可。SimCoder 程式亦可

由模擬來確認其正確性，圖 5.23 之模擬結果如圖 5.26 所示，與圖 5.18 類比之模擬接近，驗證程式之正確性。

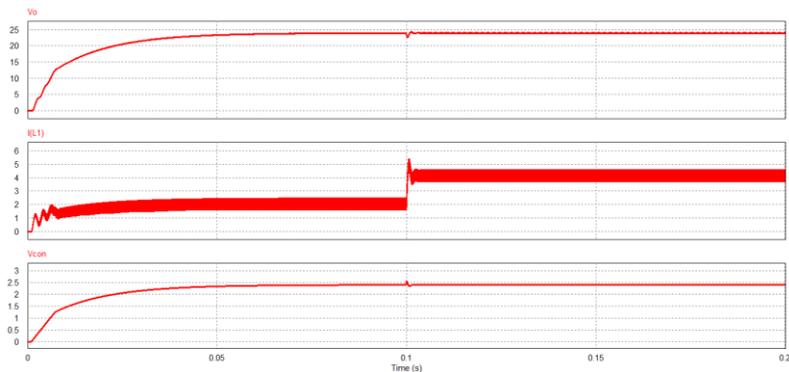


圖 5.26 (a)

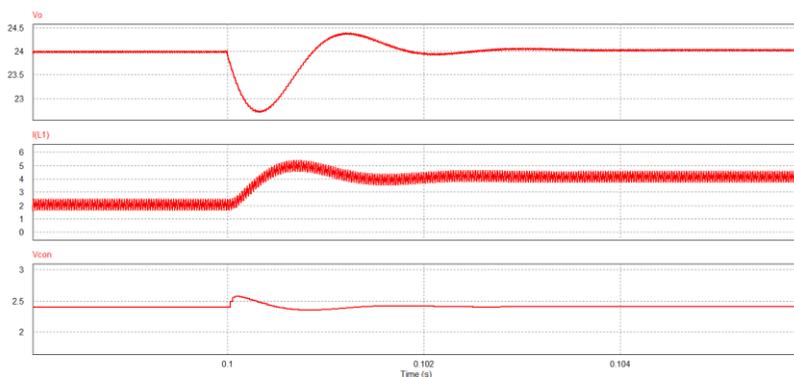


圖 5.26 (b)

圖 5.26 Buck converter 電壓模式數位控制(圖 5.21)之模擬結果具有 CV 及 CC 迴路之 SimCoder 程式如圖 5.27 所示，其可以由圖 5.21 之類比電路經由 s2z 轉換獲得。模擬結果如圖 5.28 所示，其與類比之模擬相符。

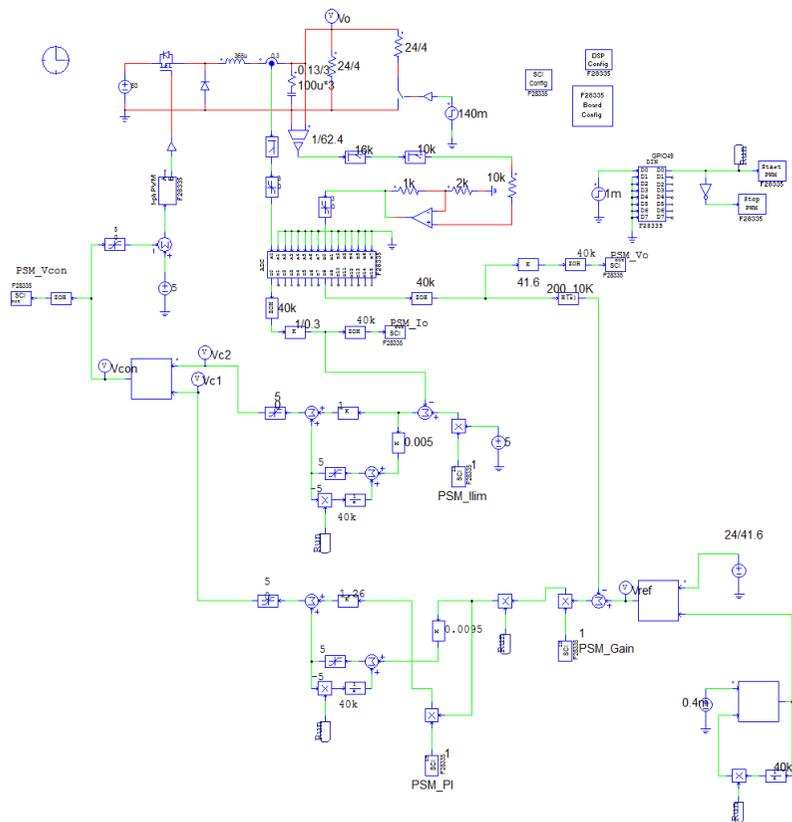


圖 5.26 (b)

圖 5.27 Buck convert 包含 CV 及 CC 控制之 SimCoder 程式

Lab2_limit.psimsch

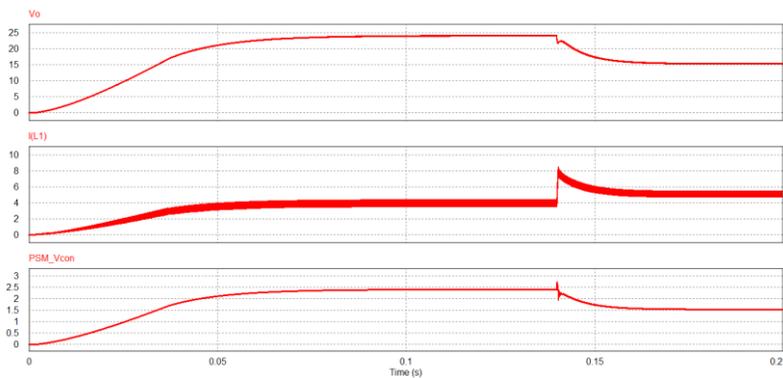


圖 5.28 圖 5.27 之模擬結果

實驗量測

實驗設備與教具配置如圖 5.29，直流電源供應器 PSW 160-7.2 連接到 Buck 教具的輸入端子 J201，其輸出端子 J202 則接至直流電子式負載 PEL-2040 並使用定電阻模式。圖 5.30 為輸出電壓與電感電流量測波形。

圖 5.29

實驗設備配置圖

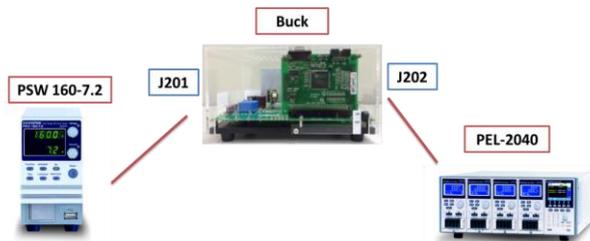
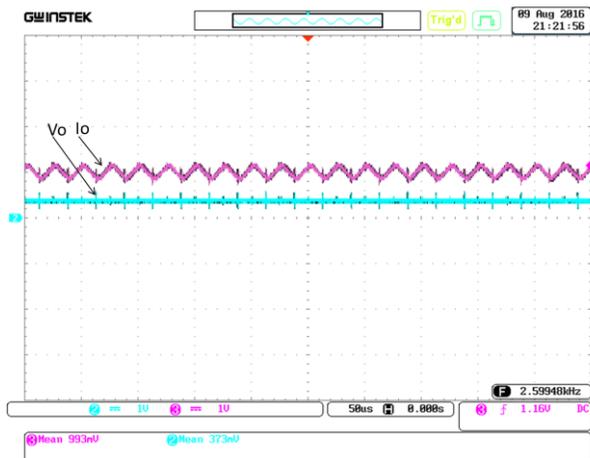


圖 5.30

轉換器輸出電壓及電感電流量測波形



實驗 3 平均電流模式控制

實驗目的

學習 Buck converter 之平均電流模式控制方法，包括電流迴路及電壓迴路之模型推導，控制器設計、硬體規劃及 SimCoder 程式撰寫等。

實驗原理

電流模式控制方法

降壓式轉換器的電流模式控制架構如圖 6.1 所示，採用雙迴路控制，外迴路為電壓迴路，用以調整電壓誤差及產生電流內迴路的電流命令 (V_c)，電流迴路命令與感測電流 ($I_L R_i$) 之誤差再經 PWM (或其他方式) 產生開關的驅動信號，感測電流可能為開關電流亦或電感電流，視採用的控制方法而定。電流模式控制的方法主要包含如圖 6.2 所示的幾種方法，包括：峰值電流控制 (peak current mode control)，波谷電流控制 (valley current mode control)，固定導通時間控制 (constant on-time control)，固定截止時間控制 (constant off-time control)，磁滯電流控制 (hysteretic current control)，平均電流控制 (average current control) 等方法，其他還有充電控制 (Charge control)、V2 固定導通時間控制 (V2 constant on-time control) 等，本實驗主要針對平均電流控制，以下說明將以此為主。

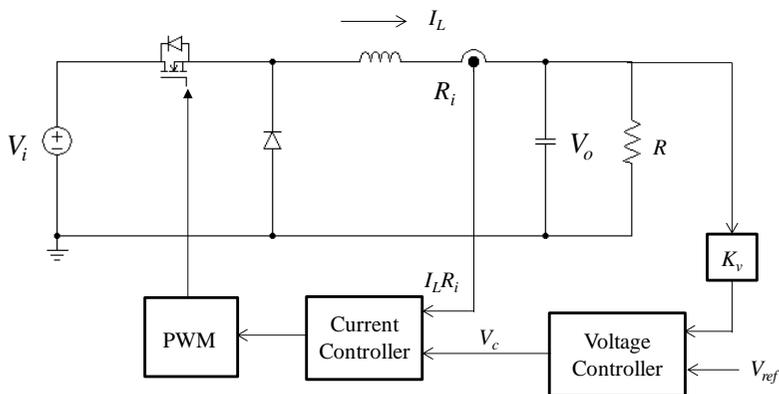
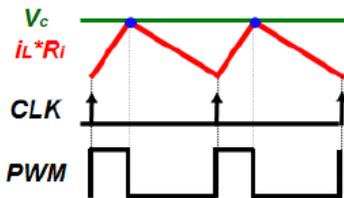


圖 6.1 電力轉換器之電流模式控制架構

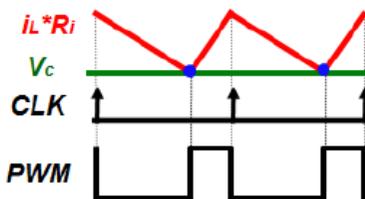
圖 6.2

電流模式控制的主要方法：

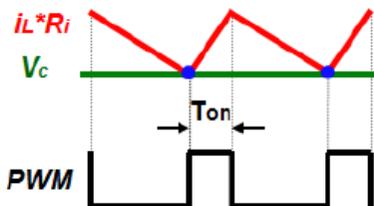
(a) 峰值電流控制



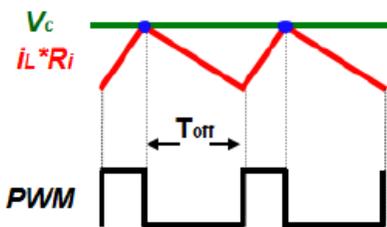
(b) 波谷電流控



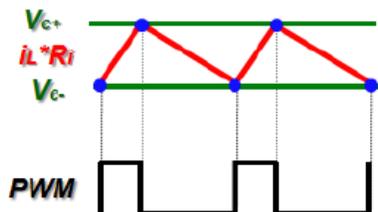
(c) 固定導通時間控制



(d) 固定截止時間
控制



(e) 磁滯電流控制

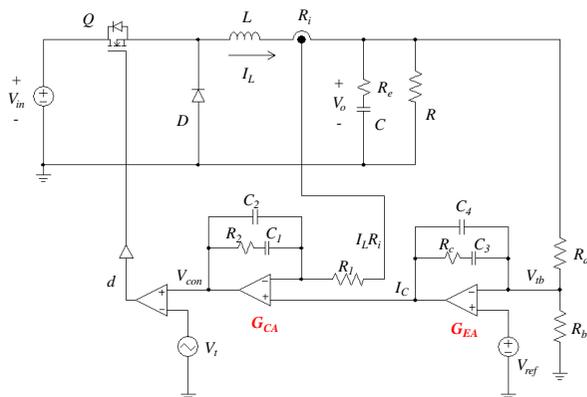


平均電流控制 (average current control) 原理

平均電流控制之控制方塊圖如圖 6.3 所示，其由外迴路電壓誤差放大器 (G_{EA}) 作電壓調整產生電感電流之命令 (I_c)，再利用感測之電感電流 ($I_L R_i$) 與 I_c 比較及經過電流之誤差放大器 (G_{CA}) 調整後產生 PWM 所需之控制電壓 (V_{con})，最後再由 V_{con} 與 PWM 之鋸齒波 (V_t) 比較得到開關之觸發信號。

圖 6.3

平均電流控制



電流迴路設計

利用狀態平均法，由圖 6.3 可得：

$$L \frac{dI_L}{dt} = dV_{in} - V_o \quad (6.1)$$

忽略 V_o 及 V_{in} 之變動，由(6.1)可得：

$$\frac{\tilde{I}_L}{\tilde{d}} = \frac{V_{in}}{sL} \quad (6.2)$$

考慮電流感測比例以及 PWM 之增益可得：

$$H_i(s) = \frac{\tilde{I}_L R_i}{\tilde{V}_{con}} = \frac{\tilde{I}_L R_i}{\tilde{d} V_t} = \frac{R_i V_{in}}{sL V_t} \quad (6.3)$$

針對一階之系統，電流誤差放大器(G_{CA})可以採用二類誤差放大器方式來設計如圖 6.4 所示，由於 PWM 之控制電壓一週期內僅能與其鋸齒波信號交會一次，因此電流迴路之最大頻寬(ω_{co})受到 V_{con} 之上升斜率小於 PWM 鋸齒波(V_t)之上升斜率之限制， V_{con} 之上升斜率可由感測之電感電流下降斜率經由 G_{CA} 放大決定，由上述限制可得：

$$(V_o / L) R_i G_{CA, \max}(\omega_{co}) = V_t f_s \quad (6.4)$$

(6.4)重新整理可得：

$$G_{CA, \max}(\omega_{co}) = \frac{\tilde{V}_{con}}{R_i \tilde{I}_L} = \frac{V_t f_s L}{V_o R_i} \quad (6.5)$$

由(6.3)及(6.5)及利用 $G_{CA, \max}(\omega_{co}) H_i(\omega_{co}) = 1$ 可得：

$$\frac{V_t f_s L}{V_o R_i} \frac{R_i V_{in}}{\omega_{co} L V_t} = 1 \quad (6.6)$$

(6.6)重新整理可得：

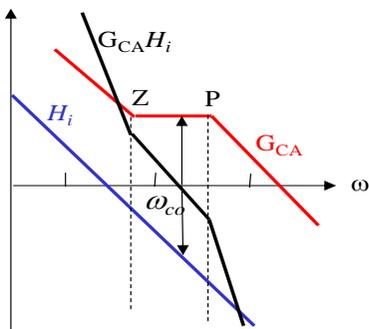
$$\omega_{co, \max} = \frac{V_{in} f_s}{V_o} \text{ (rad/s)} \quad (6.7)$$

$$f_{co, \max} = \frac{f_s}{2\pi D} \quad (6.8)$$

(6.8)指出若以控制電壓 V_{con} 上升斜率限制與來設計，理論之最高電流迴路頻寬有可能高於或太接近切換頻率，因此不可能以此值來設定，一般限於雜訊比，頻寬(ω_{co})之選擇可以設定在切換頻率的 $1/4 \sim 1/8$ 。一旦頻寬(ω_{co})選擇後，可以利用 K-factor 方法使二類誤差放大器之 $z = \omega_{co}/K$ ， $p = \omega_{co}/K$ 。

圖 6.4

平均電流控制電流迴路設計



電壓迴路設計

一般電壓迴路之響應速度遠低於電流迴路之響應速度，因此在模式化電壓迴路時，可以將電流迴路視為理想，亦即感測之電感電流與其命令之響應視為 1，即：

$$\frac{R_i \tilde{I}_L(s)}{\tilde{I}_c(s)} = 1 \tag{6.9}$$

基於此假設，平均電流控制之電壓迴路等效電路可以簡化如圖 6.5(a) 所示，由圖 6.5(a) 可得：

$$\frac{\tilde{V}_o}{\tilde{I}_L} = R \frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{\omega_o}}, \quad \omega_z = \frac{1}{CR_e}, \quad \omega_p = \frac{1}{CR} \tag{6.10}$$

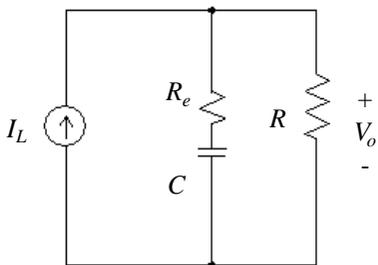
利用(6.9)及(6.10)可以繪出電壓迴路之控制方塊圖如圖 6.5(b)所示，電壓之誤差放大器(G_{EA})亦可以採用前述之二類誤差放大器來加以設計如圖 6.5(c)所示，電壓迴路之頻寬可以設計在電流迴路頻寬之 $1/3 \sim 1/5$ 。

圖 6.5

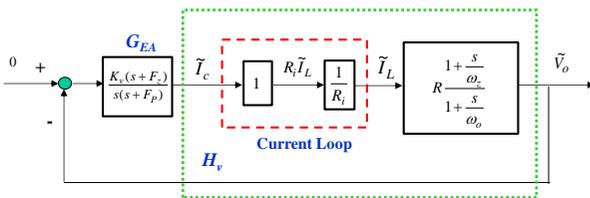
電壓誤迴路設

計：

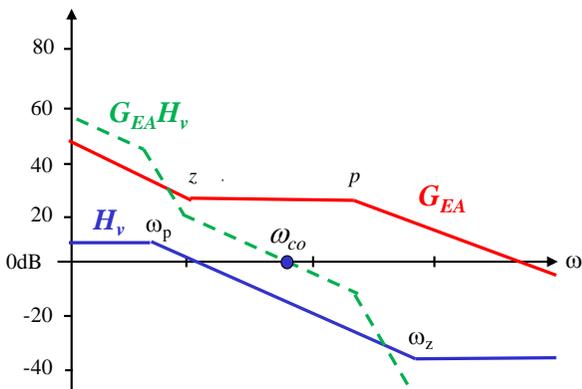
(a) 電壓迴路等效
電路



(b) 電壓迴路控制
方塊圖



(c) 電壓迴路波德
圖



本實驗平均電流控制之 Buck 轉換器規格如下，請根據上述設計方法設計其電壓及電流誤差放大器，並以模擬加以驗證：

$$V_i = 50V$$

$$V_o = 24V$$

$$f_s = 40kHz$$

$$K_s(R_i) = 0.3$$

$$K_v = 1/40$$

$$L = 365\mu H$$

$$C = 300\mu F(130m\Omega/3)$$

$$V_i = 5V$$

$$I_{o,max} = 100W/24A \quad (6.11)$$

根據上述電壓及電流迴路模型以及 2 類誤差放大器之設計方法，以 MATLAB 撰寫之設計程式如下：

■ 平均電流控制 Matlab 程式設計如下：(Buck_average.m)

% Buck with average current control

clf;

clc;

PI = 3.1416;

Vd=50;

Vo=24;

Po=100;

Io=Po/Vo;

fs=40e3;

ws = 2* PI * fs;

Kv = 1/40;

Ks = 0.3;

Ts=1/fs;

Vt=5;

Vref=Kv * Vo;

% PWM

D = Vo/Vd;

Vcon = D * Vt;

% LC

L = 365e-6;

Re = 100e-3;

C = 300e-6;

R = Vo/Io;

% Hi

num1= Ks*Vd/(Vt*L) ;

```
den1= [1 0];
Hi=tf(num1,den1);
% Gca Design - fcoi = fs/20;
% Type 2 control Gca=K(s+z)/s(s+p)
% set the z to be wcoi/5
% set the p to be LPF
fcoimax=fs/(2*PI*D);
wcoi = 2*PI*fs/20;
fcoi = wcoi/(2*PI)
z= wcoi/5
p= 2* PI * 16e3 ;
Gain1 = 1/abs(freqresp(Hi, wcoi));
numGca1= [1 z];
denGca=[1 p 0];
Gca1=tf(numGca1, denGca);
Gain2 = abs(freqresp(Gca1, wcoi));
K1 = Gain1/Gain2;
Km1 = K1/p
Gca = K1 * Gca1;
GHi=Gca*Hi;
GHio = freqresp(GHi, wcoi);
phaseGHio = angle(GHio) * 180/PI;
PMi = 180 + phaseGHio
figure(1);
bode(Hi, Gca, GHi, {100 5000000});
grid;
% Hv
wz = 1/(Re*C);
wp = 1/(R*C);
num2= R*[1/wz 1];
```

```
den2= [1/wp 0];
Hv1=tf(num2, den2);
p1 = 16e3 * 2 * PI
p2 = 10e3 * 2 * PI
numFB1 = Kv * p1;
denFB1 = [1 p1];
FB1=tf(numFB1, denFB1);
numFB2 = p2;
denFB2 = [1 p2];
FB2=tf(numFB2, denFB2);
FB = series(FB1, FB2);
Hv = series(Hv1, FB);
% PI control G=Km2(s+z)/s
fcov = fcoi/20
wcov=2*PI*fcov;
z= wcov/5
tu = 1/z
Gain3 = 1/abs(freqresp(Hv, wcov));
numGea1= [1 z];
denGea=[1 0];
Gea1=tf(numGea1, denGea);
Gain4 = abs(freqresp(Gea1, wcoi));
Km2 = Gain3/Gain4
Gea = Km2 * Gea1;
GHv=Gea*Hv;
GHvo = freqresp(GHv,wcov);
phaseGHvo = angle(GHvo) * 180/PI;
PMv = 180 + phaseGHvo
figure(2);
bode(Hv, Gea, GHv, {100 5000000});
```

grid;

■ 設計結果如下：

$$f_{coi} = 2000$$

$$z = 2.5133e+003$$

$$K_{m1} = 1.5109$$

$$PM_i = 71.5653$$

$$p_1 = 1.0053e+005$$

$$p_2 = 62832$$

$$f_{cov} = 100$$

$$z = 125.6640$$

$$t_u = 0.0080$$

$$K_{m2} = 7.5386$$

$$PM_v = 78.8391$$

(6.12)

以上採用平均電流控制之控制迴路波德圖如圖 6.6 所示，圖 6.6(a)所示為電流迴路，圖 6.6(b)則為電壓迴路。

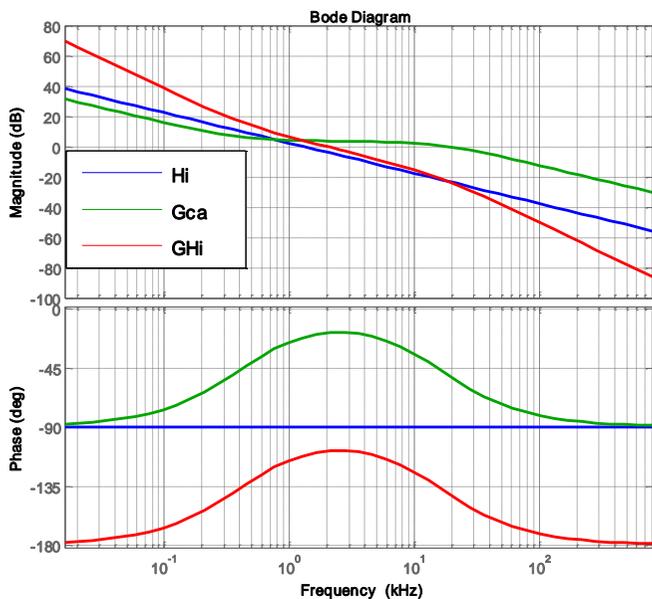


圖 6.6 (a) 電流迴路

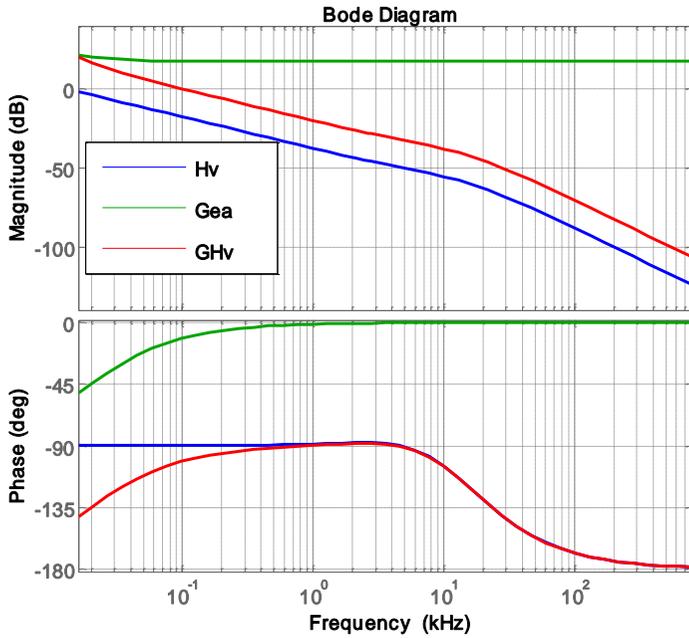


圖 6.6 (b) 電壓迴路

圖 6.6 採用平均電流控制之控制迴路波德圖

電路模擬

依據上述設計所建制之 PSIM 模擬電路如圖 6.7 所示，模擬結果則如圖 6.8 所示。

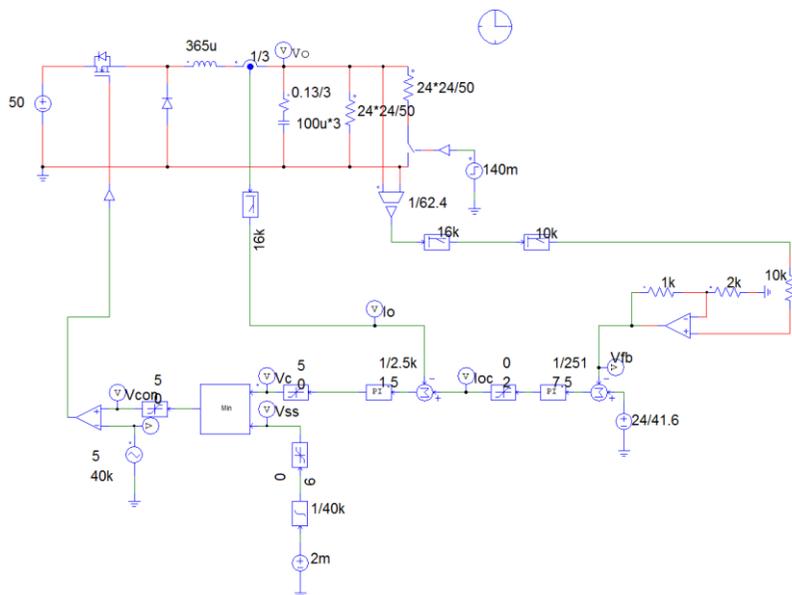


圖 6.7 PSIM 模擬電路(buck3.psim)

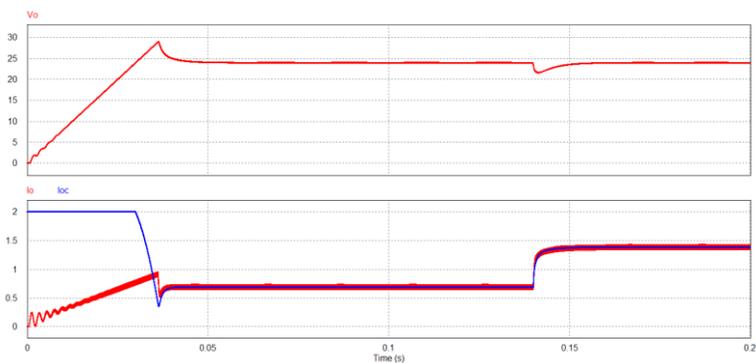


圖 6.8 圖 6.7 之模擬結果

SimCoder 程式規劃

利用 SimCoder 建置之程式如圖 6.9 所示，其模擬結果如圖 6.10 所示，均與類比模擬相符。

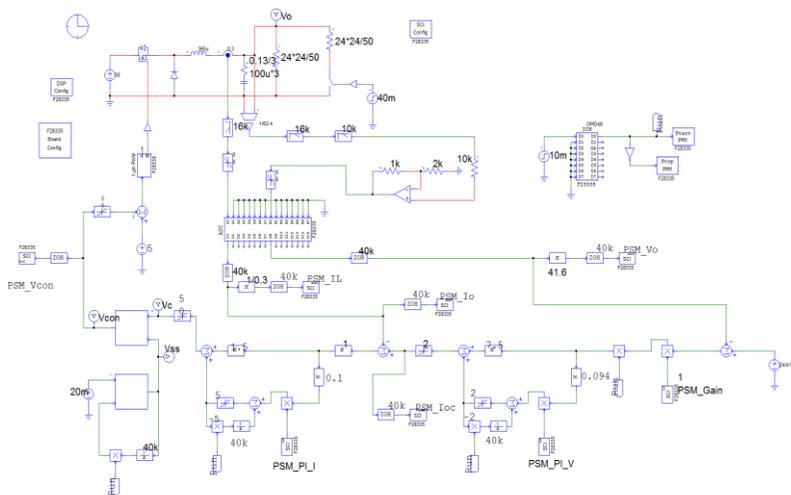


圖 6.9 SimCoder 程式(Lab3.psimsch)

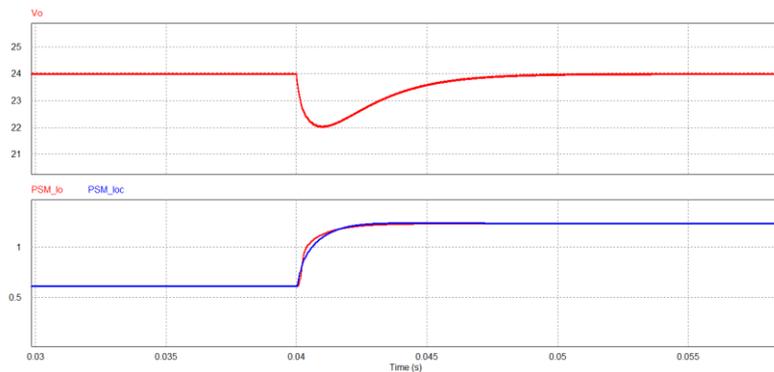


圖 6.10 圖 6.9 之模擬結果

實驗量測

實驗設備與教具配置如圖 6.11，直流電源供應器 PSW 160-7.2 連接到 Buck 教具的輸入端子 J201，其輸出端子 J202 則接至直流電子式負載 PEL-2040 並使用定電阻模式。

圖 6.11

實驗設備配置圖

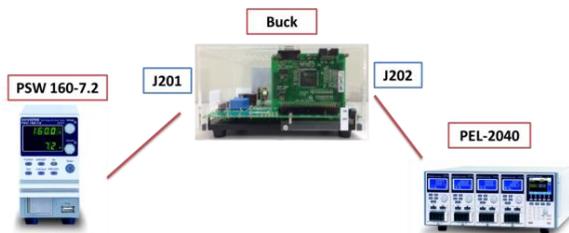
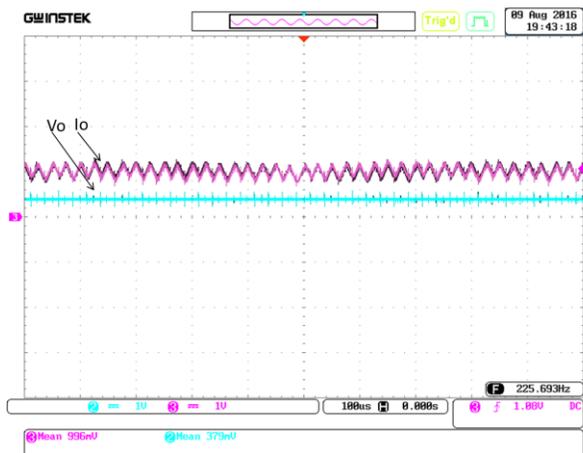


圖 6.12

轉換器輸出電壓
及電感電流之量
測波形



實驗 4 最大功率點追蹤控制

實驗目的

學習 PV 模組之最大功率點追蹤控制方法、包括擾動觀察法、增量電導法等常用之 MPPT 控制方法、硬體規劃及其控制程式之撰寫等。

實驗原理

PV 模組之特性介紹

光伏(photovoltaic, PV)模組的輸出特性既非定電壓亦或定電流，輸出功率均會隨工作點改變，因此必須隨時調整工作點，使發電功率達到最佳，此稱為最大功率點追蹤(maximum power point tracking, MPPT)，以下針對 PV 模組特性加以說明，以作為 MPPT 控制器設計之參考。

PV 模組乃由許多 PV 串並聯所組成，PV 經由光照射後，形成一電流源提供負載能量，PV 等效電路如圖 4.1 所示，其中 I_{ph} 用來表示 PV 所產生之電流，而 D_j 則表示為一 P-N 界面二極體， R_{sh} 與 R_s 則分別表示材料內部之等效並聯與串聯電阻，通常在分析時 R_{sh} 的值很大，而 R_s 的值則很小，因此一般而言，為了簡化分析過程可將 R_{sh} 和 R_s 忽略不計， I_p 與 V_p 分別表示 PV 之輸出電流與電壓。

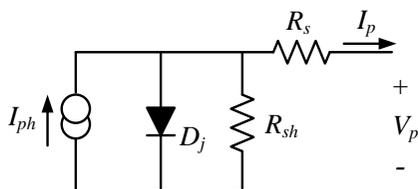
PV 模組的等效電路和 P-N 界面半導體的特性，PV 模組的輸出電流可用以下數學方程式(7.1)來表示

$$I_{pv} = n_p I_{ph} - n_p I_{sat} \left[\exp \left(\frac{q}{kTA} \frac{V_{pv}}{n_s} \right) - 1 \right] \quad (7.1)$$

- V_{pb} : 表示 PV 之輸出電壓(V)
- I_{pv} : 表示 PV 之輸出電流(A)
- T : 表示 PV 之表面溫度(°K)
- A : 表示 PV 之理想因數($A=1\sim5$)
- q : 表示一個電子之電荷量($1.6\times 10^{-19}C$)
- k : 表示波茲曼常數($1.38\times 10^{-23} J/^{\circ}K$)
- n_p : 表示 PV 之電池並聯數
- n_s : 表示 PV 之電池串聯數

圖 7.1

PV 之等效電路



而 I_{sat} 表示太陽光電板的反向飽和電流，其數學關係式可表示如下：

$$I_{sat} = I_{rr} \left(\frac{T}{T_r}\right)^3 \cdot \exp\left[\frac{qE_{gap}}{kA} \left(\frac{1}{T_r} - \frac{1}{T}\right)\right] \quad (7.2)$$

- T_r : 表示 PV 之參考溫度(°K)
- I_{rr} : 表示 PV 之參考溫度 T_r 時之反向飽和電流(A)
- E_{gap} : 表示半導體材料跨越能間帶間隙時所需能量

$$E_g = 1.16 - 7.02 \times 10^{-4} \frac{T^2}{T - 1108} \quad (7.3)$$

$$I_{ph} = [I_{scr} + \alpha(T - T_r)] \frac{S}{100}$$

- I_{scr} : 表示 PV 電池工作於參考溫度與 $1kW/m^2$ 之日照條件下之短路電流(A)
- α : 為 PV 模組短路電流溫度係數
- S : 為太陽日照量(kW/m^2)

由式(7.3)可計算出輸出功率 P，表示為

$$P_{pv} = V_{pv} \times I_{pv} \quad (7.4)$$

將式(7.1)與式(7.3)乘開後，可得到

$$P_{pv} = n_p V_{pv} I_{ph} - n_p I_{sat} V_{ph} \left[\exp\left(\frac{q}{kT} \frac{V_{pv}}{n_s}\right) - 1 \right] \quad (7.5)$$

由上述方程式(7.5)可了解 PV 模組特性，可描繪出在不同日照量與 PV 板表面溫度變化情況下 PV 模組輸出電壓、電流與功率曲線圖。

以一 Shell 製造之 75W PV 模組如表 7.1 所示。

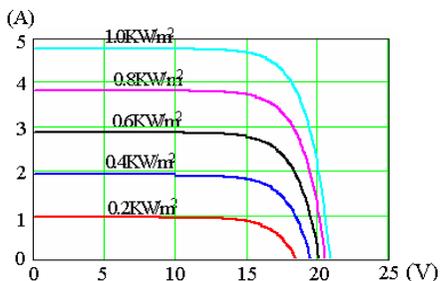
電氣特性	規格
額定最大輸出功率(W)	75
額定電流(A)	4.4
額定電壓(V)	17.0
短路電流 I_{sc} (A)	4.8
開路電壓 V_{oc} (V)	21.7
正常工作溫度 NOTC($^{\circ}$ C)	45.2
短路電流溫度係數 K_i (mA/ $^{\circ}$ C)	2.06
開路電壓溫度係數(V/ $^{\circ}$ C)	-0.77

表 7.1 Shell SQ75 PV 規格表

在溫度 25° C 與太陽日照量為表 $1\text{KW}/\text{m}^2$ 時，其電氣特性為表 7.1 所示，在不同日照量下輸出電壓與電流及輸出電壓與功率曲線分別為圖 7.2(a)與圖 7.2(b)，及在不同溫度下輸出電壓與電流及輸出電壓與功率曲線分別為圖 7.3(a)與圖 7.3(b)。

圖 7.2

不同照度、固定
環境溫度(25° C)
下
(a) I_p - V_p 特性曲
線



(b) $P-V_p$ 特性曲線

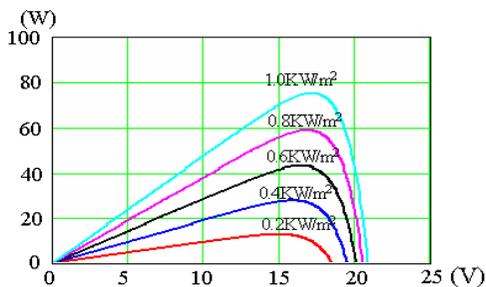
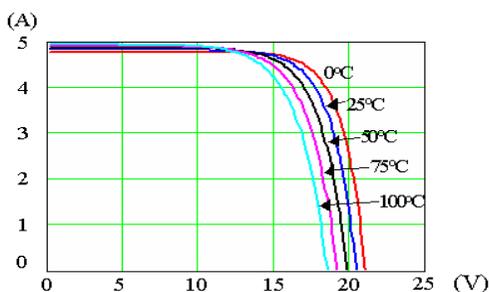
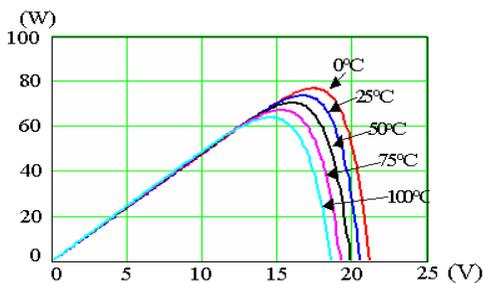


圖 7.3

不同環境溫度、
固定照度
(1kW/m^2) 下 (a)
 I_p-V_p 特性曲線



(b) $P-V_p$ 特性曲線



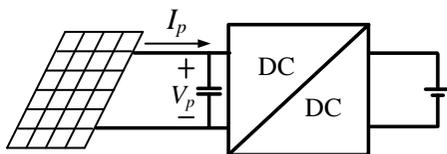
PV 電轉換器之控制方法

由 PV 模組之特性模擬曲線可得知，影響 PV 模組輸出功率的兩個因素為日照強度與環境溫度，隨著天氣的多變，溫度與日照強度隨時都可能改變，因此欲使 PV 模組輸出最大功率，提高 PV 模組發電效率必須控制 PV 系統的功率轉換器，來獲得在不同工作環境下最大功率輸出，而此控制方法即是最大功率點追蹤(MPPT)技術。

圖 7.4 為三種典型 PV 發電系統之電力轉換器架構，圖 7.4(a)為一蓄電池充電系統，圖 7.4(b)與(c)為市電並聯系統，三種架構之 MPPT 控制都必須藉由控制 PV 模組轉換器之輸入電壓 V_p 與輸入電流 I_p 來達成，只要計算出 PV 模組最大功率點之電壓做為命令，藉由電壓迴路控制使其操作在 MTPP 點即可。

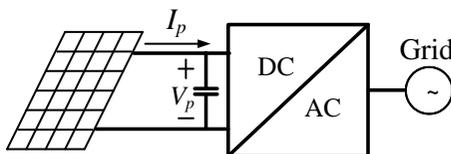
圖 7.4

各式 PV 模組電力轉換器系統架構

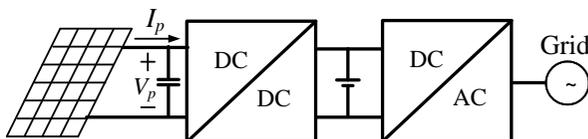


(a) 蓄電池充電系統

(b) 單級市電並聯系統



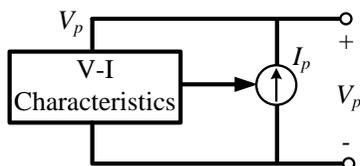
(c) 雙級市電並聯系統



由 PV 模組之特性模擬曲線可知，PV 模組既非電壓源亦非電流源，當應用於電壓源轉換器結合分析時，例如圖 7.4 中之各式轉換器時，則 PV 模組必須視為一電流源，其等效電路如圖 7.5(a)所示；反之亦然，當應用於電流源轉換器時，則 PV 模組必須視為一電壓源，其等效電路如圖 7.5(b)所示。

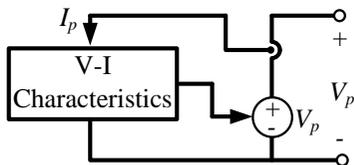
圖 7.5

PV 模組之等效電路



(a) 針對電壓源轉換器

(b) 針對電流源轉換器



MPPT 方法

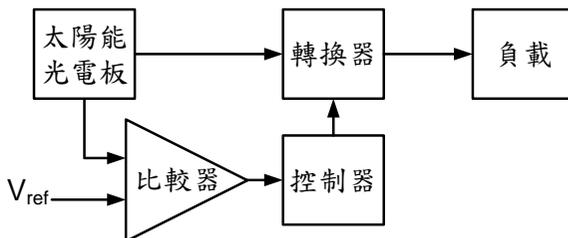
針對 MPPT 方法，過去有許多種方法被提出，包括電壓回授法、功率回授法、擾動與觀察法、增量電導法、直線近似法與實際量測法等六類，以上幾種最大功率追蹤法，其基本概念是相同的。皆是利用 PV 輸出電壓與電流，或是兩者都使用來做最大功率追蹤法，主要差別是在於最大功率點的判斷與實現不同。各式 MPPT 控制方法說明如下：

(1) 電壓回授法

電壓回授法中是最簡單的一種方法，如圖 7.6 所示，原理則是在已知日照與溫度下最大功率點下電壓大小，藉由調整 PV 模組輸出電壓來達到最大功率追蹤的目的，而主要缺點為當環境改變時系統不能自動追蹤至新的最大功率點，造成能量損耗。

圖 7.6

電壓回授法方塊圖



(2) 功率回授法

功率回授法是與電壓回授法類似，如圖 7.7 所示，由於電壓回授法無法在變動的大氣條件下，追蹤至新的最大功率點，因此在功率回授法中加入輸出功率對電壓的變化率判斷式，以達到應變大氣條件變化時能最追蹤至最大功率點。相較電壓回授法雖然多了電壓變化率的判斷式但也因此降低了能量損失，在提昇效率上是比電壓回授法好。

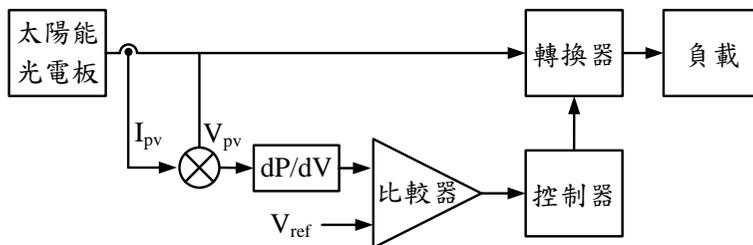


圖 7.7 功率迴授法方塊圖

(3) 擾動觀察法

擾動觀察法，如圖 7.8 所示，此結構簡單，只需要量測 PV 模組輸出電壓與電流，與功率迴授法一樣，所以被普遍應用在 PV 模組最大功率追蹤上。

擾動觀察法的基本原理是藉由週期性之增加或減少負載之大小，來改變 PV 模組之輸出電壓與功率，也就是改變 PV 板於特性曲線上之工作點，並觀察與比較負載變動前後輸出電壓與輸出功率之大小再決定下次增減載的動作。若擾動造成 PV 模組輸出功率較前次增加，即於下一周期適度地往同方向增加或減少負載，使輸出功率不斷增加；反之亦然，在輸出功率此變動前小的情況下，則表示需要在下一週期改變負載變動方向。如此反覆擾動與觀察比較，能使 PV 模組達到其最大功率點上，即是擾動觀察法的動作原理。若欲響應速度快，但電壓之晃動越大，因此其準確性與響應速度需作一折衷。

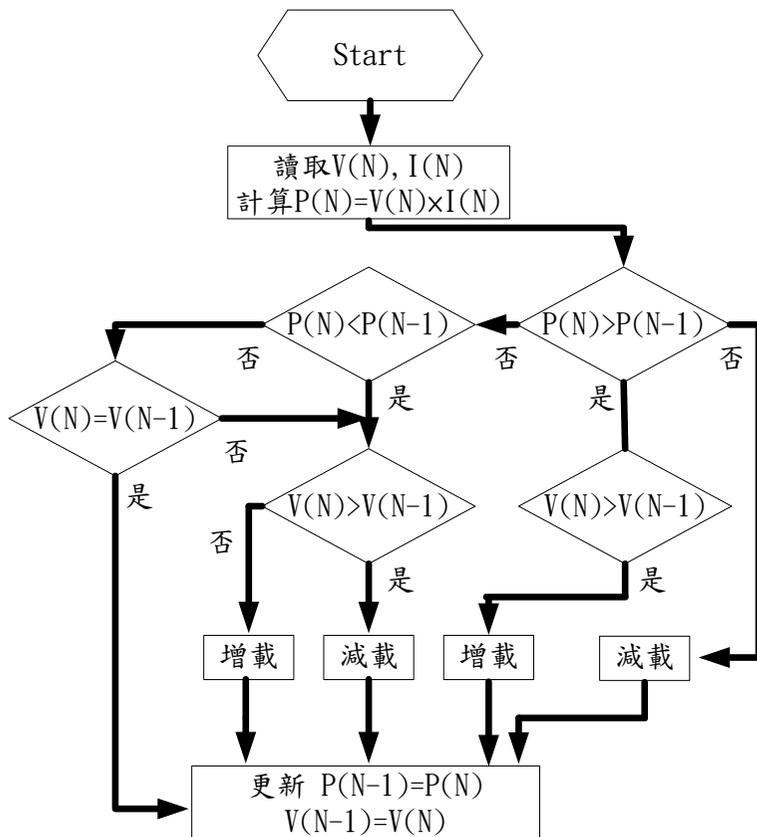


圖 7.8 擾動與觀察法之控制方塊圖

(4) 增量電導法(incremental conductance, INC)

如圖 7.9 所示，增量電導法的概念與功率迴授法相同，主要是利用

$dP_{pv}/dV = 0$ 判斷式，將式子改寫為

$$\frac{dP_{pv}}{dV_{pv}} = \frac{d(I_{pv}V_{pv})}{dV} = I + V \frac{dI}{dV} = 0 \tag{7.9}$$

整理式(7.9)可得

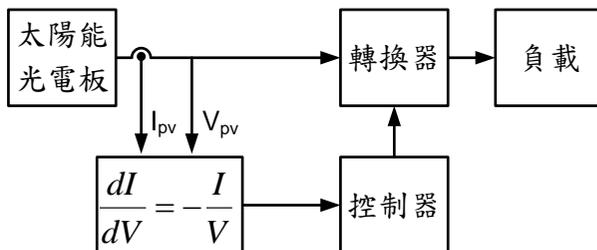
$$\frac{dI}{dV} = -\frac{I}{V} \quad (7.10)$$

式(7.10)中， dI 是表示增量前後所量到的電流變化量，而 dV 是表示增量前後電壓變化量。藉由量測增量值 dI/dV 與瞬時 PV 的電導 I/V 可以決定下一次的變動量，當增量值與電導做符合左右兩邊相等的關係時，則表示已達到最大功率點而不進行下一次擾動，此即是增量電導法基本原理。

雖然增量電導法是以改變 PV 模組輸出電壓來達到最大功率追蹤來達到最大功率點，不同於擾動觀察法的判斷邏輯，可避免掉如擾動觀察法在最大功率追蹤附近擺動的問題。

圖 7.9

增量電導法方塊圖

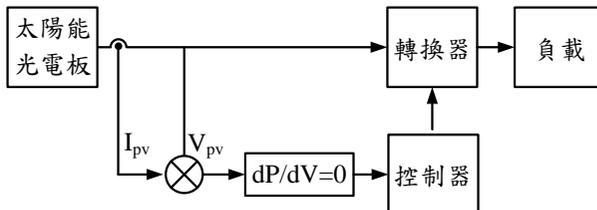


(5) 直線近似法

直線近似法以圖 7.10 所示，直線近似法的基本概念在於利用 $dP_{pv}/dV = 0$ 這個邏輯判斷式上，並以一直線來近似 PV 模組在固定溫度下種不同日照量的最大功率點，將 PV 模組輸出電流控制在此直線上，即可達到最大功率追蹤，以數學模型為基礎來推導 PV 模組的最大功率點上的近似直線，在採用此法時各項參數的正確性與元件老化會迫使此直線近似法失去準確性。

圖 7.10

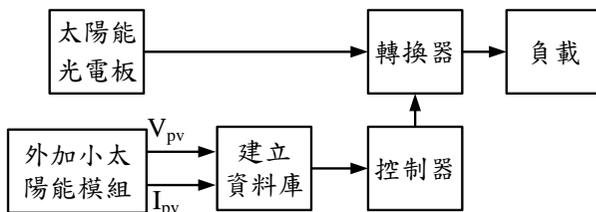
直線近似法方塊圖



(6) 實際量測法

實際量測法如圖 7.11 所示，主要是利用額外的 PV 模組，每隔一段時間去量測 PV 板的開路電壓與短路電流，建立在此大氣條件下的日照與溫度其最大功率點時之電壓與電流，搭配控制電路使 PV 模組工作在此電壓與電流下，即可獲得精準的最大功率點，但此情況只針對氣候變化不大之地區，若環境改變則需重新量測建立資料庫為主要缺點。

圖 7.11
實際量測法方塊圖



最大功率追蹤方法的工作原理及優缺點比較整理如表 7.2 所示。

MPPT 方法	工作原理	優點	缺點
電壓迴授法	事先量測得 PV 板特性為參考	架構簡單、成本低	不能自動追蹤至新的最大功率點
功率迴授法	與電壓迴授相似，增加輸出功率對電壓變化判斷	減少能量損耗增加整體效率	與電壓迴授法相比，計算量較大
擾動觀察法	週期增減負載，觀察功率與電壓變化，決定下一步增減負載	實現容易、架構與原理皆簡單	在最大功率點上會有擺動功率損失
增量電導法	利用 PV dI/dV 與 I/V 間關係來決定增量值	相較擾動觀察法減少了振盪功率損失	量測精密度要很高，實際使用時誤差大
直線近似法	用 $dP/dV=0$ 並以一直線來近似 PV 最大功率點	實現容易、架構容易	PV 板與元件老化失去準確性

實際量測法	外接一 PV 板量測其特性、建立參考模型而控制	可避免 PV 老化而失去模型精準度	環境改變則需重新量測建立資料庫
-------	-------------------------	-------------------	-----------------

表 7.2 最大功率追蹤方法的工作原理及優缺點比較

(7) 改良型增量電導(INC) MPPT 控制法

本實驗將 PV 模組的電壓電流輸出曲線視同等效模型如圖 4.12 所示為隨著照度變化的電壓源 V_S 串聯一內阻 R_S ，提供能量給 DC-DC 轉換器，此處之 DC-DC 轉換器形同一負載 R_L ，在此模型下 PV 模組於不同負載下的輸出功率 P_o 可寫成：

$$P_o = V_o \times I_o = V_o \times \frac{V_S - V_o}{R_S} = \frac{V_o \cdot V_S - V_o^2}{R_S} \quad (7.11)$$

V_o 為輸出電壓， I_o 為輸出電流。此電壓-功率輸出關係為圖 4.13 的曲線 a ，其最大功率點發生在：

$$\frac{dP_o}{dV_o} = \frac{V_S}{R_S} - 2 \frac{V_o}{R_S} = 0 \Rightarrow V_o = \frac{V_S}{2} \quad (7.12)$$

因此，當 PV 模組輸出電壓為 $V_S/2$ 時，具有最大功率輸出。再根據分壓定理：

$$V_o = V_S \left(\frac{R_L}{R_S + R_L} \right) = \frac{V_S}{2} \Rightarrow R_L = R_S \quad (7.13)$$

證明當 $R_L = R_S$ 時，PV 模組具最大功率輸出，也就是阻抗匹配定理。實際上 PV 模組之 V_s 與 R_s 值如同前述圖 4.2 及圖 4.3 之曲線可知，雖然在同日照下此二值將然隨工作點變化而動態變化，因此 R_L 值亦會動態變化，然而上述二電阻分壓之最大功率值理論仍然適用。

圖 7.12

PV 模組等效電路模型

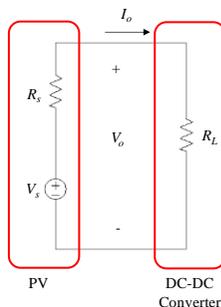


圖 7.13

PV 模組電壓、
電流與阻抗之關係圖

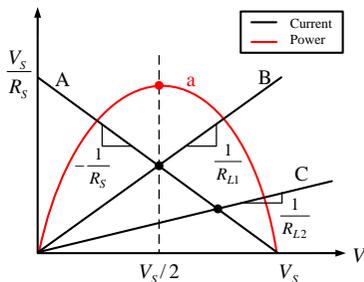


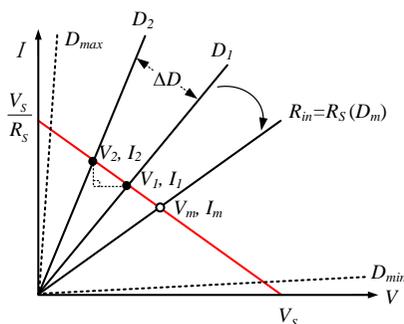
圖 7.14 所示為本實驗所採用之 PV 模組 MPPT 方法，利用操作於兩個負載值時得到的電壓電流，利用兩點求斜率法，可求得 PV 模組內阻，如圖 4.14 所示。此方法先操作 DC-DC 轉換器於不同的責任週期率(D)，等同於兩種不同的負載值，操作於第一個導通率 D1 時感測輸入的電壓電流 V_1, I_1 ，再操作第二個導通率 D_2 (與 D_1 間格 ΔD) 為 V_2, I_2 ，得到兩點後再經由下式：

$$R_S = -\frac{V_1 - V_2}{I_1 - I_2} \tag{7.14}$$

可得到 PV 模組的內阻 R_S ；也可以量測多個電壓電流值後，以線性迴歸方法計算得內阻值。之後量測輸入電壓與電流值，計算得輸入阻抗 R_{in} ，調整導通率以改變輸入阻抗，並使其等同於內阻 R_S ，或者與內阻 R_S 相差在某個微小值之內。當程式執行一段時間後 ($t > T$, t 為已執行時間, T 為設定之重覆執行時間)，將會重新量測輸入阻抗，最終達到 $R_S = R_L$ ，亦即最大功率點。若將 $1/R_S$ 視為增量電導，則此種阻抗匹配法，其實等效於上述之 INC MPPT 方法。

圖 7.14

PV 之阻抗匹配
MPPT 方法

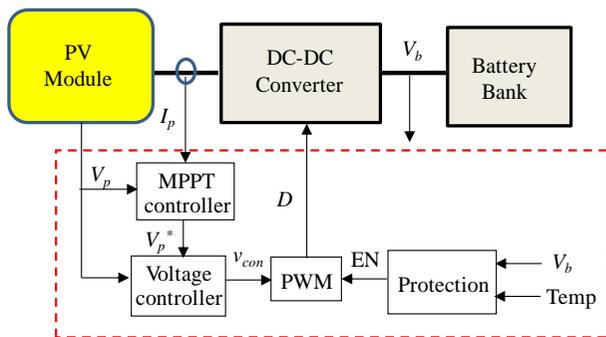


雖然圖 7.14 之方法可以追蹤最大功率點，然而由於責任週期改變後 DC-DC 轉換器必需經歷一段時間電壓達到穩態後再繼續下一點之計

算，才不致有方向誤判之情況，因此 MPPT 速度將受到電路開迴路響應速率之限制。為提昇 MPPT 速度，必須縮短工作點改變後達到穩態之時間，為達成此目標，本研究 DC-DC 轉換器之控制架構如圖 7.15 所示，採用雙迴路控制，外迴路為 MPPT 控制器，其感測輸入電壓及電流作 MPPT 控制用以產生 PV 模組之電壓命令(V_p^*)。內迴路為電壓控制迴路，其利用電壓命令與迴授之輸入電壓(V_p)相較經誤差放大器調整後得到開關之責任週期。具有電壓迴路之控制架構，其電壓調整後達到穩態之時間較直接改變責任週期電壓達到穩態之時間要短許多，因此可以改善響應速度，並使 MPPT 之追蹤更精確。

圖 7.15

DC-DC 轉換器之控制架構



MPPT 之追蹤方法仍然使用上述之電阻匹配法，僅將工作點之調整方式改為電壓，其程式流程如圖 4.15 所示。程式首先設定起始值之電壓命令為開路電壓，緊接進入週期為 T 之迴圈並進行取樣 PV 模組之電壓及電流(V_1 及 I_1)，接著改變其電壓命令得到(V_2^*)，此處*代表命令，此電壓命令改變值在起始時預設為 $-\Delta V$ ，但下一週期 T 之後則可能為 $+\Delta V$ 亦或 $-\Delta V$ ，視其前次取樣電壓改變值而定，但改變後之 V_2^* 若大於 PV 模組預設之電壓最高值 V_{max} 則增加值改為 $-\Delta V$ ，同理改變後之 V_2^* 若小於 PV 模組預設之電壓最低值 V_{min} 則增加值改為 $+\Delta V$ 。接著取樣 V_2 及 I_2 ，再利用(V_1, I_1)及(V_2, I_2)計算 R_s 值。其後進入週期時間 ΔT 之迴圈，不斷利用感測之(V_1, I_1)計算 R_L 值，再由 R_s 值與 R_L 之比較結果調整電壓命令，當 $R_s \geq R_L$ 電壓之調整為增加 ΔV 以加大 R_L ，反之當 $R_s < R_L$ 電壓之調整為減少 ΔV 以減低 R_L ，使 R_L 得以追蹤 R_s 。此程序反覆執行，直到達到一時間周期 T 後，進入下一個時間 T 之迴圈。

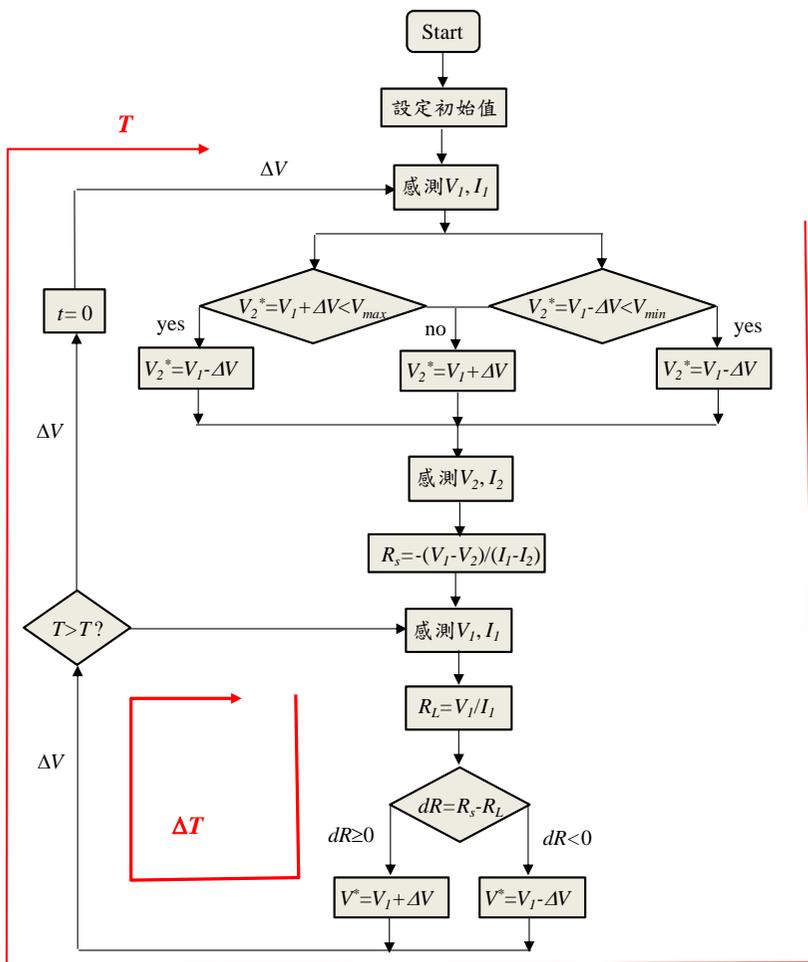


圖 7.16 採用雙迴路控制電阻匹配法之 MPPT 方法程式流程

電路模擬

根據上述 MPPT 控制方法及前面實驗所設計之 Buck converter，以 PSIM 電路建置之 P&O MPPT 控制電路如圖 7.17 所示，其 MPPT 程式如圖 7.17 下所附之程式所示，MPPT 程序之 $\Delta T=1/200\text{Hz}$ 。模擬結果如圖 7.18 所示，其確實能夠準確快速的追蹤到最大功率點 ($V_{mp}=32\text{V}$, $I_{mp}=3\text{A}$, $P_{mp}=96\text{W}$)。

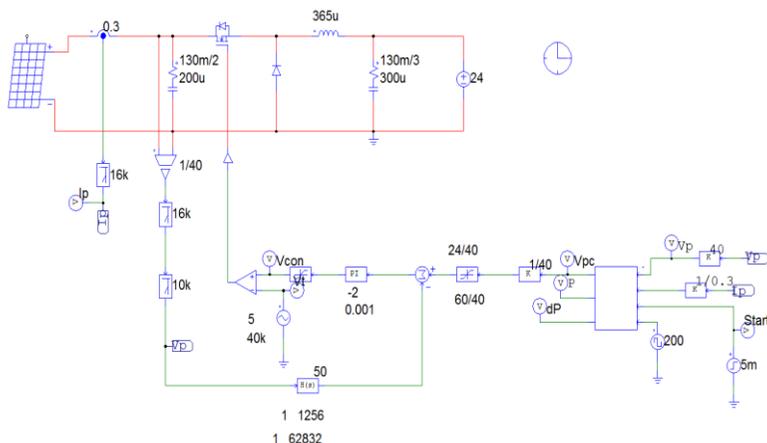


圖 7.16 採用 P&O MPPT 控制法之模擬電路(buck_4_P&O.psimsch)

P&O MPPT 程式 (圖 7.16)

```

static int n;
static double ymax = 50;
static double ymin = 24;
static double Vp, Ip, Vp1, Ip1, P, P1, dP, dv=1, Vpc, Vpc1, start, CLK,
CLK1;
start = x3;
Vp = x1;
Ip = x2;
CLK = x4;
P = Vp * Ip;
if (start <1)
{
Vpc = Vp - dv;
CLK1 = CLK;
}
if (start >0)

```

```

{
  if((CLK-CLK1)==1)
  {
    if (P > P1)
    {
      if (Vp > Vp1)
        { Vpc = Vp + dv; }
      else
        { Vpc = Vp - dv; }
    }
  }
else
  {
    if (Vp > Vp1)
      { Vpc = Vp - dv; }
    else
      { Vpc = Vp + dv; }
  }
  if (Vpc >= ymax)
    { Vpc = Vp - dv; }
  if (Vpc <= ymin)
    { Vpc = Vp + 3 * dv; }
  Vp1 = Vp;
  Vpc1 = Vpc;
  dP = P - P1;
  P1 = P;
}
CLK1 = CLK;
}
y1 = Vpc;
y2 = P;
y3 = dP;

```

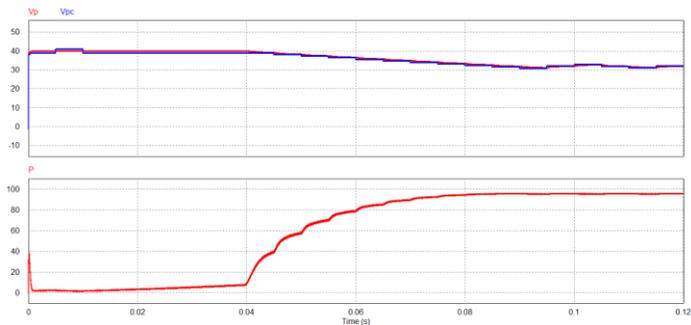


圖 7.17 圖 7.16 之模擬結果

以 PSIM 電路建置之 INC MPPT 控制電路如圖 7.18 所示，其 MPPT 程式如圖 7.18 下所附之程式所示，MPPT 程序之頻率為 200Hz，模擬結果如圖 7.19 所示，其 R_L 與 R_s 確能快速的緊密追隨，因此亦能夠準確快速的追隨到最大功率點($V_{mp}=32V, I_{mp}=3A, P_{mp}=96W$)。

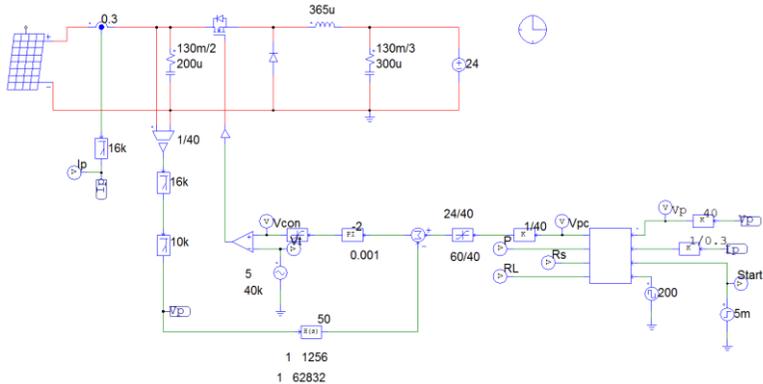


圖 7.18 採用 INC MPPT 控制法之模擬電路(buck_4_INC.psimsch)

INC MPPT 程式 (圖 7.18)

```

static double ymin = 24;
static double Vp, Ip, Vp1, Ip1, P, P1, Rs, RL, dV, dI, dv=1, Vpc, Vpc1,
start, CLK, CLK1;
start = x3;
Vp = x1;
Ip = x2;
CLK = x4;
P = Vp * Ip;
if (start <1)
{
Vpc = Vp - dv;
CLK1 = CLK;
}
if (start >0)
{
if((CLK-CLK1)==1)
{
dV = Vp - Vp1;
dI = Ip - Ip1;
Rs = dV/dI;
if (Rs < 0)

```

```

    { Rs = -Rs;
      if (Rs >= 2000)
        {Rs = 2000;}
    }
    RL = Vp/Ip;
    if (RL >= 5000)
      {RL = 5000;}
    if (Rs > RL)
      { Vpc = Vp + dv;}
    if (Rs <= RL)
      { Vpc = Vp - dv;}
      if (Vpc >= ymax)
        { Vpc = Vp - dv;}
        if (Vpc <= ymin)
          { Vpc = Vp + 3 * dv;}
    Vp1 = Vp;
    Vpc1 = Vpc;
    Ip1 = Ip;
  }
  CLK1 = CLK;
}
y1=Vpc;
y2 = P;
y3 = Rs;
y4 = RL;

```

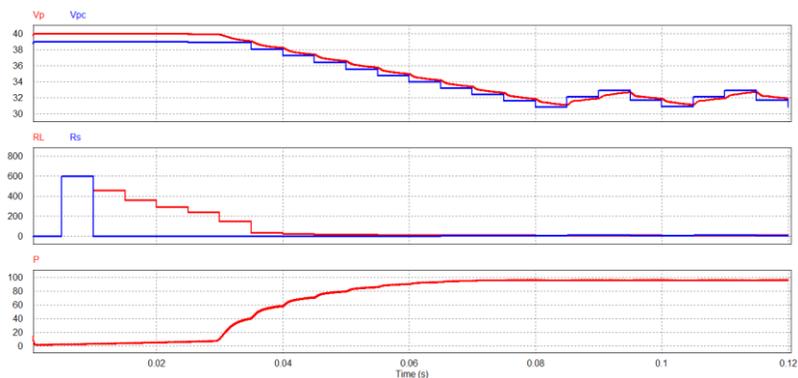


圖 7.19 圖 7.18 之模擬結果

SimCoder 程式規劃

圖 7.20 所示為以 SimCoder 電路建置之 P&O MPPT 控制程式，其乃根據圖 7.16 電路作數位化，MPPT 程式如圖 7.20 下所附之程式所示，MPPT 程序為 200Hz。模擬結果如圖 7.21 所示，其與類比程式之模擬結果(圖 7.17)一樣，確實能夠準確快速的追蹤到最大功率點 ($V_{mp}=32V$, $I_{mp}=3A$, $P_{mp}=96W$)。在此要注意的是，圖 7.20 僅供模擬，因其 200Hz 之 MPPT 調整頻率對於一般 PV 模擬器之響應太快，實際上實作之程式如圖 7.22 所示，MPPT 之調整頻率改為 1Hz，此頻率太慢了，要由模擬來確認響應較為困難，因此圖 7.22 僅供實作。其 MPPT 程式亦顯示於圖 7.22 下，與 MPPT 200Hz 之程式相比，略作調整，將 MPPT 調整之步距電壓隨功率之變化而調整以降低 MPPT 點附近之晃動。

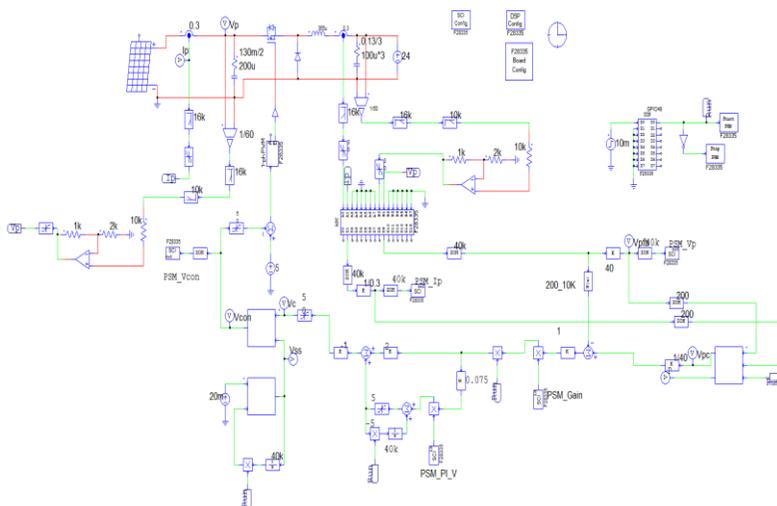


圖 7.20 以 SimCoder 電路建置之 P&O MPPT 控制程式 (Lab4_P&O.psimsch, 僅供模擬)

P&O MPPT 程式 (圖 7.20)

```
static double ymax = 50;
static double ymin = 24;
static double Vp, Ip, Vp1, Ip1, P, P1, dv=1, Vpc, Vpc1, start;
start = x3;
Vp = x1;
```

```
Ip = x2;
P = Vp * Ip;
if (start <1)
{
Vpc = Vp - dv;
}
if (start >0)
{
    if (P >=P1)
    {
        if (Vp > Vp1)
        { Vpc = Vp + dv; }
        else
        { Vpc = Vp - dv; }
    }
    else
    {
        if (Vp > Vp1)
        { Vpc = Vp - dv; }
        else
        { Vpc = Vp + dv; }
    }
}
if (Vpc>=ymax)
{ Vpc= Vp - 2 * dv;}
if (Vpc<=ymin)
{ Vpc= Vp + 3 * dv;}
P1 = P;
Vp1 = Vp;
Ip1 = Ip;
}
y1=Vpc;
y2 = P;
```

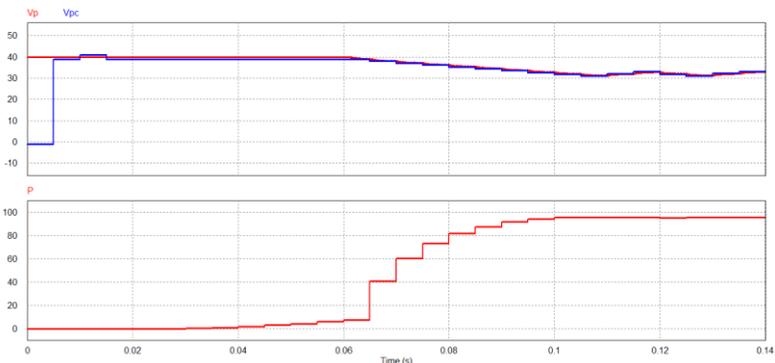


圖 7.21 圖 7.20 之模擬結果

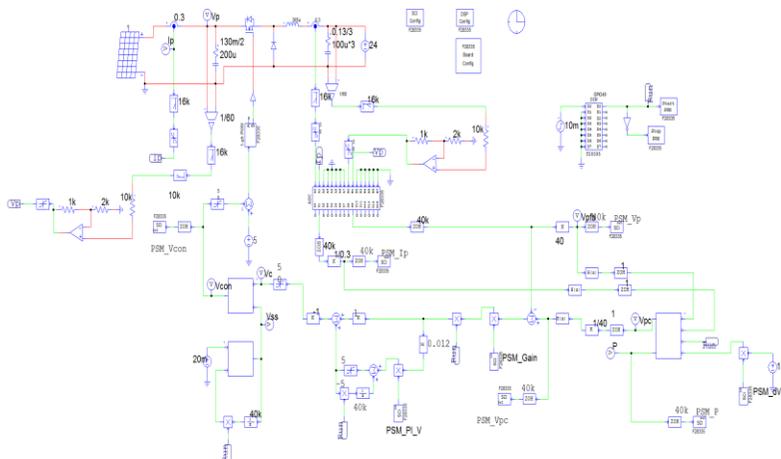


圖 7.22 以 SimCoder 電路建置之 P&O MPPT 控制程式 (Lab4_P&O_R.psimsch, 僅供實作)

P&O MPPT 程式 (圖 7.22)

```

static double ymax = 50;
static double ymin = 24;
static double Vp, Ip, Vp1, Ip1, P, P1, dP, dv, Vpc, Vpc1, start;
start = x3;
Vp = x1;
Ip = x2;
P = Vp * Ip;
dv = x4;
    
```

```
if (start <1)
{
Vpc = Vp - dv;
}
if (start >0)
{
dP=P-P1;
if (dP<0)
{dP = -dP;}
if (dP>5)
{dv = 0.5;}
if ((dP<=5)&&(dP>2))
{dv = 0.25;}
if (dP<=2)
{dv = 0.1;}
if (P >=P1)
{
if (Vp > Vp1)
{ Vpc = Vp + dv; }
else
{ Vpc = Vp - dv; }
}
else
{
if (Vp > Vp1)
{ Vpc = Vp - dv; }
else
{ Vpc = Vp + dv; }
}
if (P<3)
{Vpc = Vp - 1;}
if (Vpc>=ymax)
{ Vpc= Vp - 2 * dv;}
if (Vpc<=ymin)
{ Vpc= Vp + 3 * dv;}
P1 = P;
Vp1 = Vp;
Ip1 = Ip;
}
y1=Vpc;
y2 = P;
```

圖 7.23 所示為以 SimCoder 電路建置之 INC MPPT 控制程式，其乃根據圖 7.18 電路作數位化，MPPT 程式如圖 7.23 下所附之程式所示，MPPT 程式為 200Hz。模擬結果如圖 7.24 所示，其與類比程式之模擬結果(圖 7.19)一樣，確實能夠準確快速的追蹤到最大功率點 ($V_{mp}=32V$, $I_{mp}=3A$, $P_{mp}=96W$)。在此要注意的是，圖 7.23 僅供模擬，因其 200Hz 之 MPPT 調整頻率對於一般 PV 模擬器之響應太快，實際上實作之程式如圖 7.25 所示，MPPT 之調整頻率改為 1Hz，此頻率太慢了，要由模擬來確認響應較為困難，因此圖 7.25 僅供實作。其 MPPT 程式亦顯示於圖 7.25 下，與 MPPT 200Hz 之程式相比，略作調整，將 MPPT 調整之步距電壓隨功率之變化而調整以降低 MPPT 點附近之晃動。

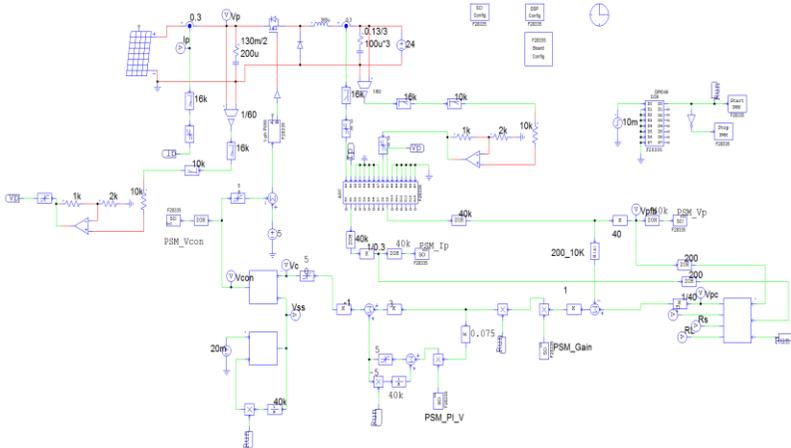


圖 7.23 以 SimCoder 電路建置之 INC MPPT 控制程式
(Lab4_INC.psim.sch，僅供模擬)

INC MPPT 程式 (圖 7.23)

```
static double ymax = 50;
static double ymin = 24;
static double Vp, Ip, Vp1, Ip1, P, dv=1, Vpc, Vpc1, dV, dI, Rs, RL, start;
start = x3;
Vp = x1;
Ip = x2;
P = Vp * Ip;
if (start < 1)
{
```

```
Vpc = Vp - dv;  
Ip1 = Ip - 0.01;  
}  
if (start >0)  
{  
    dV = Vp - Vp1;  
    dI = Ip - Ip1;  
    Rs = dV/dI;  
    if (Rs < 0)  
        { Rs = -Rs;  
          if (Rs >= 2000)  
              {Rs = 2000;}  
          }  
    RL = Vp/Ip;  
    if (RL >= 5000)  
        {RL = 5000;}  
    if (Rs > RL)  
        { Vpc = Vp + dv;}  
    if (Rs <= RL)  
        { Vpc = Vp - dv;}  
    if (Vpc >= ymax)  
        { Vpc = Vp - 2 * dv;}  
    if (Vpc <= ymin)  
        { Vpc = Vp + 3 * dv;}  
    Vp1 = Vp;  
    Vpc1 = Vpc;  
    Ip1 = Ip;  
}  
y1 = Vpc;  
y2 = P;  
y3 = Rs;  
y4 = RL;
```

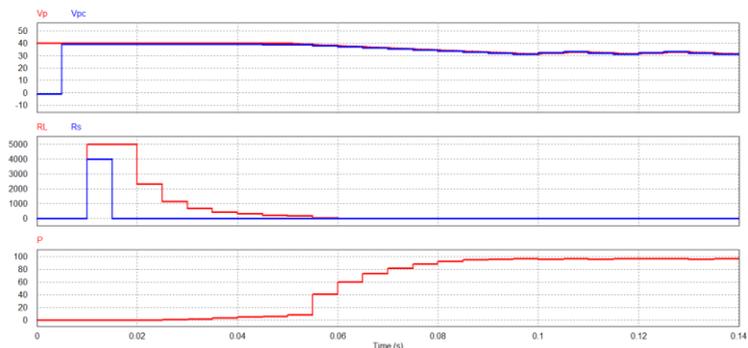


圖 7.24 以 7.23 之模擬結果

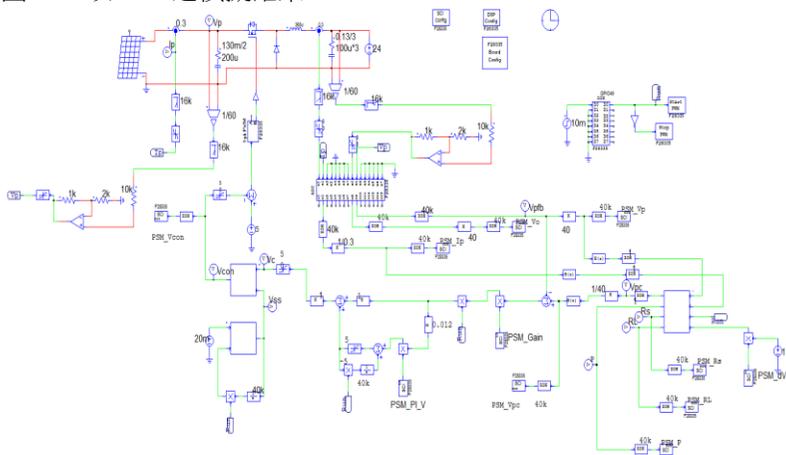


圖 7.25 以 SimCoder 電路建置之 INC MPPT 控制程式

(Lab4_INC_R.psimsch, 僅供實作)

INC MPPT 程式 (圖 7.25)

```

static double ymax = 50;
static double ymin = 24;
static double Vp, Ip, Vp1, Ip1, P, P1, dP, dv, Vpc, Vpc1, dV, dI, Rs, RL, Rs1,
Rs2, start;
start = x3;
Vp = x1;
Ip = x2;
dv = x4;
P = Vp * Ip;
if (start < 1)
    
```

```

{
Vpc = Vp - dv;
Ip1 = Ip - 0.01;
}
if (start >0)
{
dP=P-P1;
if (dP<0)
{dP = -dP;}
if (dP>5)
{dv = 0.75;}
if ((dP<=5)&&(dP>2))
{dv = 0.3;}
if (dP<=2)
{dv = 0.2;}
dV = Vp - Vp1;
dI = Ip - Ip1;
if (dI<0)
{dI = -dI;}
if (dI<0.001)
{dI = 0.001;}
Rs = dV/dI;
if (Rs < 0)
{ Rs = -Rs;
if (Rs >= 2000)
{Rs = 2000;}
}
Rs = (Rs2 + Rs1 + Rs)/3;
RL = Vp/Ip;
if (RL >= 5000)
{RL = 5000;}
if (Rs> RL)
{ Vpc = Vp + dv;}
if (Rs <= RL)
{ Vpc = Vp - dv;}
if (P<3)
{Vpc = Vp - 1;}
if (Vpc>=ymax)
{ Vpc= Vp - 2 * dv;}
if (Vpc<=ymin)
{ Vpc= Vp + dv;}
Vp1 = Vp;
Vpc1 = Vpc;
Ip1 = Ip;

```

```
Rs2 = Rs1;  
Rs1 = Rs;  
P1 = P;  
}  
y1=Vpc;  
y2 = P;  
y3 = Rs;  
y4 = RL;
```

實驗量測

實驗設備與教具配置如圖 7.26，直流電源供應器 PSW 160-7.2 連接到 Buck 教具的輸入端子 J201，其輸出端子 J202 則接至直流電子式負載 PEL-2040 並使用定電壓模式。圖 7.27 為以 Labview 呈現之 MPPT 的 P-V 與 I-V 曲線。

圖 7.26
實驗設備配置圖

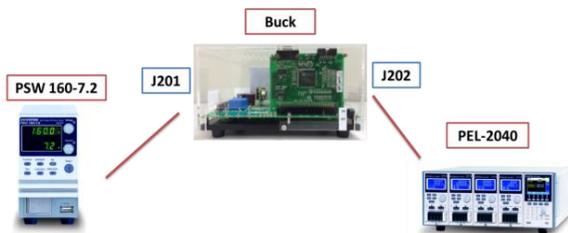
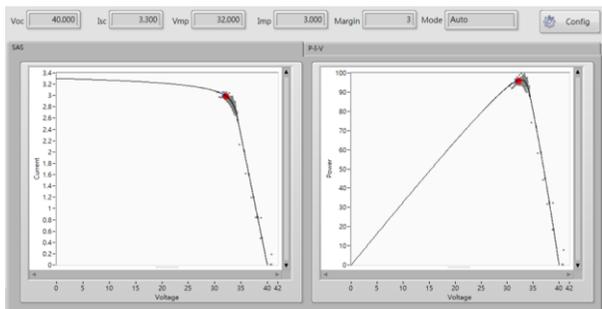


圖 7.27
MPPT 之 P-V 與
I-V 曲線



實驗 5 PV 充電器

實驗目的

學習 PV 充電器之控制方法、包括電池三階段充電控制方法、MPPT 控制與充電控制方法之結合、硬體規劃及其控制程式之撰寫等。

實驗原理

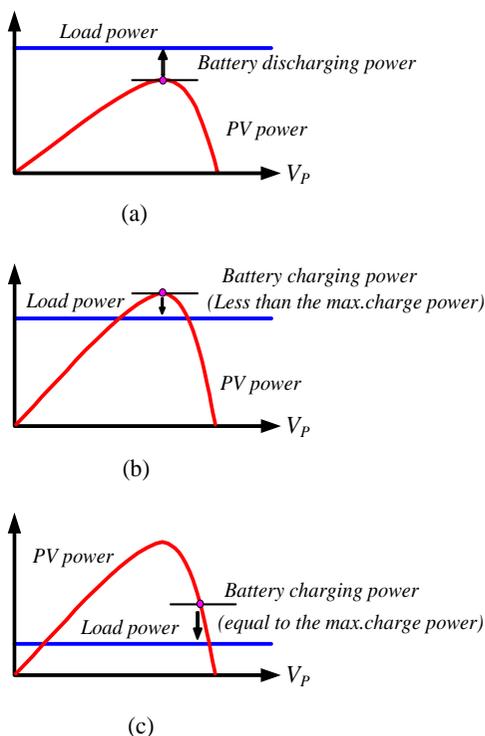
PV 充電系統工作模式

圖 8.1 所示為 PV 充電系統之三種工作模式，圖 8.1(a)為 MPPT 模式，當 PV 模組電力不足供給負載或充電使用時，其不足部分將由蓄電池供給；圖 8.1(b)為部分充電模式，當 PV 模組電力足夠供給負載使用，且多餘之電力還可對蓄電池進行充電，但所多餘電力仍不足以最大充電電流對蓄電池充電，以上兩種模式均操作於最大功率點。圖 8.1(c)為 MPPT 偏移模式，當 PV 模組操作於最大功率點時，其總輸出電力大於負載及蓄電池需求總和，為求電力平衡情況下，必須降低 PV 模組總輸出電力，此時 PV 模組必須偏離其最大功率點以降低所產生之電力，來達到電力平衡。

圖 8.1

PV 充電系統工作模式

- (a) MPPT 模式
(b) 部分充電模式
(c) MPPT 偏移模式



控制架構

為實現上述工作模式，PV 充電系統控制架構安排如圖 8.2 所示，其採用雙迴路控制，外迴路為充電控制(*charge control*)迴路，內迴路包含 MPPT 控制迴路與 PV 電壓控制(*PV control*)迴路，其控制原理如下：

首先取樣蓄電池電壓(V_b)與蓄電池最大電壓命令(V_b^*)比較，其誤差經由 PI 控制器與正向限制器後可得到充電電流命令(I_b^*)，當蓄電池電壓未達到最大電壓命令時，此限制器用以設定最大充電電流；當蓄電池電壓接近於最大電壓命令時，控制器便進入線性區，使得充電電流命令減少，最終將達到電池最大電壓命令準位且使充電電流僅提供電池本身之消耗。此方法可達到定電流充電、定電壓充電及浮充等三階段充電之目的來增長電池壽命。

由 PI 電壓控制器產生之充電電流命令(I_b^*)與蓄電池電流(I_b)比較，當 $I_b > I_b^*$ 時，其誤差經由 PI 控制器與正向為零之限制器後產生一電壓命令(V_L)，此電壓命令(V_L)將會影響其最大功率點，使得最大功率點偏向右半面曲線，提升 PV 模組輸出電壓，最終使 PV 模組所產生輸出功率為負載和蓄電池需求總合，以達到電力平衡，此情形發生於 MPPT 偏移模式。反之，當 $I_b < I_b^*$ 時，其誤差經 PI 控制器和正向為零限制器後為零，因此充電迴路便無法影響 MPPT 所產生之電壓命令(V_p^*)，即太陽能輸出電壓命令乃由 MPPT 控制器決定，因此 MPPT 模式和部分充電模式之 PV 模組均操作於最大功率點狀態。

依據上述方法，本實驗電池三階段充電策略之設定如下，其中 $I_{ch,max}$ 乃根據電池形式及容量所設定之最大充電電流：

定電流區: $V_b \leq 0.8V_b^*$

$$I_b^* = I_{ch,max}$$

定電壓區: $0.8 V_b^* < V_b \leq 0.95V_b^*$

$$I_b^* = I_{ch,max} \sim 0.1 I_{ch,max} \text{ (視電池電壓線性調整)}$$

浮充區: $0.95 V_b^* < V_b \leq V_b^*$,

$$I_b^* = 0.1 I_{ch,max}$$

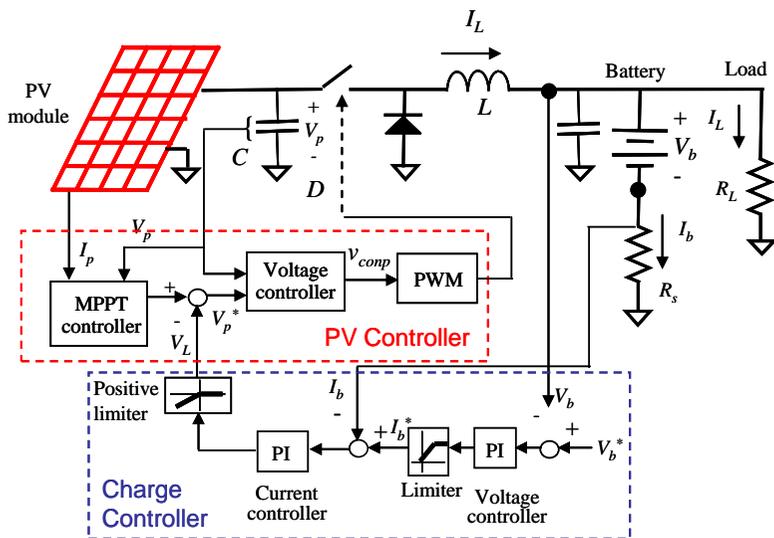


圖 8.2PV 充電系統架構圖

電路模擬

結合 PV 控制器(MPPT 採用 P&O 擾動觀察法)與充電控制器之模擬電路如圖 8.3 所示, PV 模組之最大功率點設定為 $V_{mp}=32V$, $I_{mp}=3A$, $P_{mp}=96W$ 。電池充電器之設定為 $V_b^*=28V$, 電池電壓為 $24V$, 當充電電流限制(I_b^*)設定為 $5A$ 時, 由於在此設定下輸出功率($P_{ch}=24V*5A=120W$)超過 P_{mp} , 系統將操作於輸出將被限制在 P_{mp} 使 PV 模組操作於最大功率點, 亦即前述之 MPPT 模式(如果有接上負載的話)亦或部分充電模式, 模擬結果如圖 8.4 所示, 系統確實操作於 MPP 點。電池電壓為 $24V$, 當充電電流限制(I_b^*)設定為 $3A$ 時, 由於在此設定下輸出功率($P_{ch}=24V*3A=72W$)低於 P_{mp} , 系統之輸出將由 P_{ch} 決定, MPPT 將產生偏移, 亦即前述之 MPPT 偏移模式, 模擬結果如圖 8.5 所示, MPP 點確實偏移, 且系統之輸出由 P_{ch} 決定。將輸出之電池電壓提高到 $28V$, 充電控制器將進入 CV 模式, 使 I_b^* 減少, MPPT 將更加偏移, PV 模組接近開路電壓, 模擬結果如圖 8.6 所示。

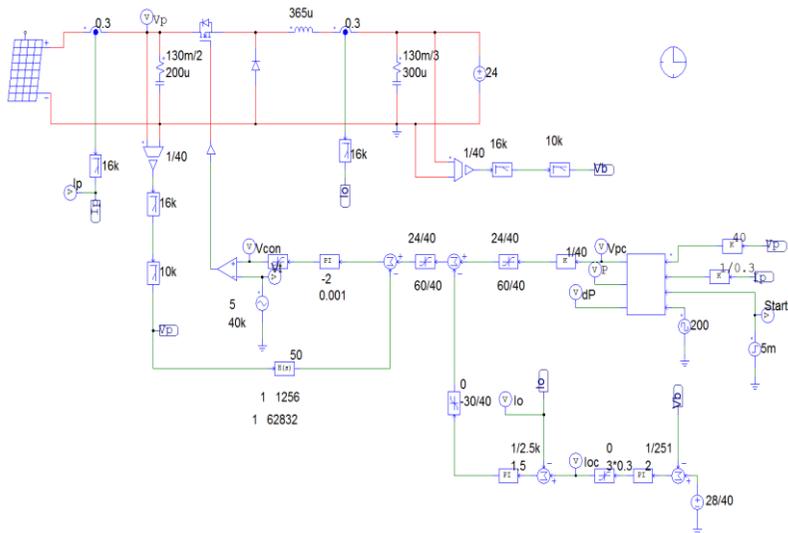


圖 8.3 採用 P&O MPPT 之 PV 充電器 (buck_5_P&O_charge.psimsch)

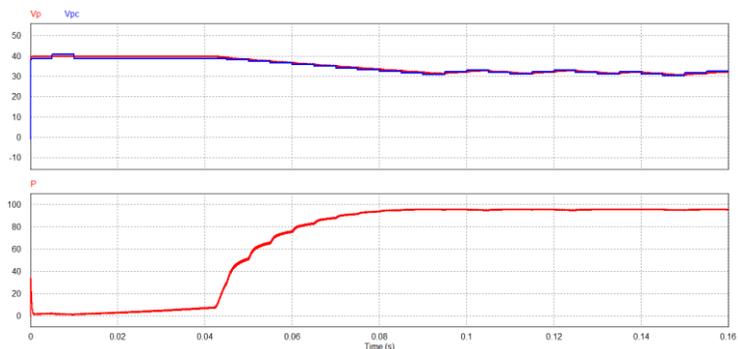


圖 8.4 充電控制器操作於 CC 且發電功率達不到 CC 電流設定之需求 (MPPT 模式亦或部分充電模式)

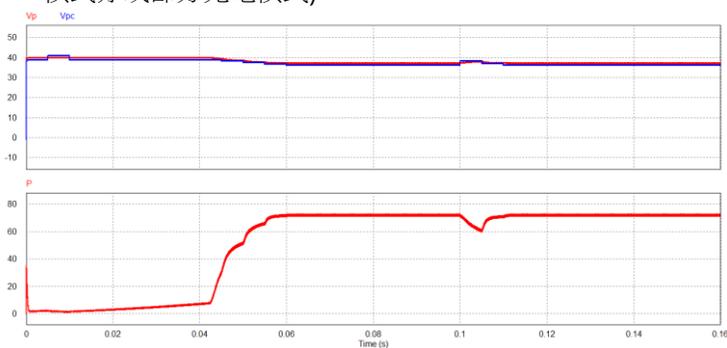


圖 8.5 充電控制器操作於 CC 且可發電功率高於 CC 電流設定之需求 (MPPT 偏移模式)

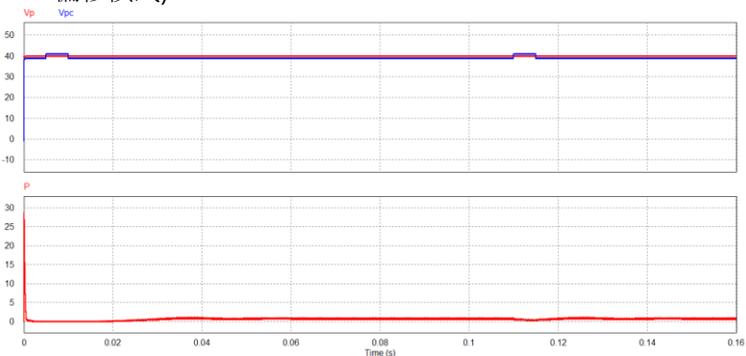


圖 8.6 充電控制器操作於 CV 且可發電功率高於 CV 電流設定之需求 (MPPT 偏移模式)

SimCoder 程式規劃

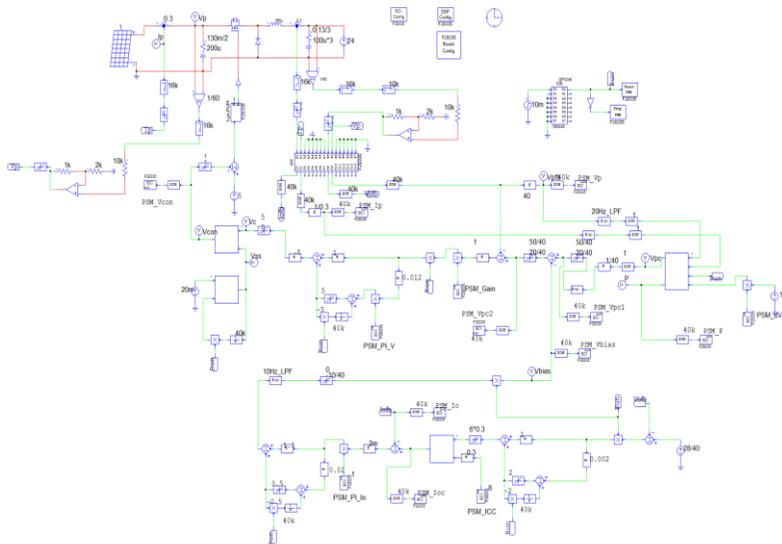


圖 8.8 採用 P&O MPPT 之 PV 充電器 SimCoder 程式
(Lab5_P&O_charge_R)

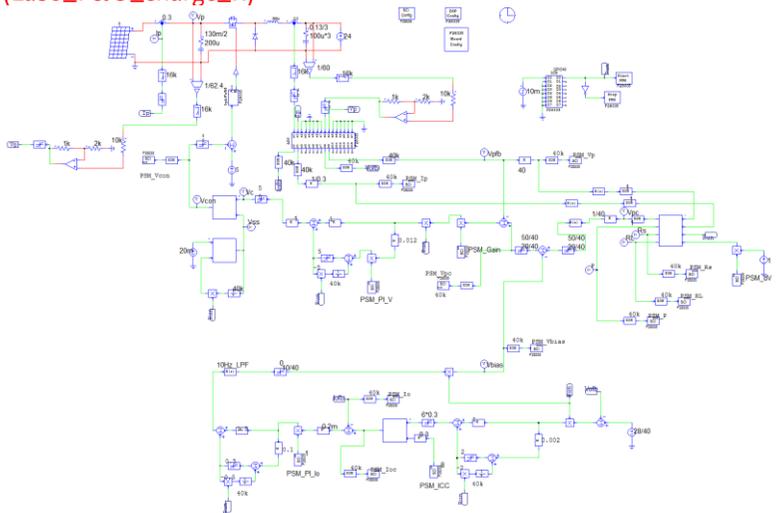


圖 8.9 採用 INC MPPT 之 PV 充電器 SimCoder 程式
(Lab5_INC_charge_R)

實驗量測

實驗設備與教具配置如圖 8.10，直流電源供應器 PSW 160-7.2 連接到 Buck 教具的輸入端子 J201，其輸出端子 J202 則接至直流電子式負載 PEL-2040 並使用定電壓模式。

圖 8.10

實驗設備配置圖

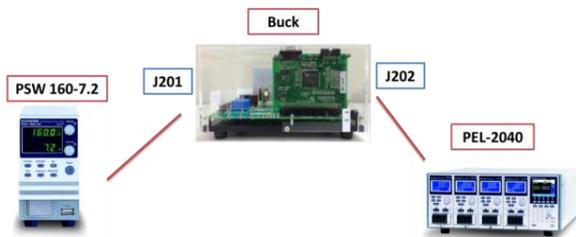
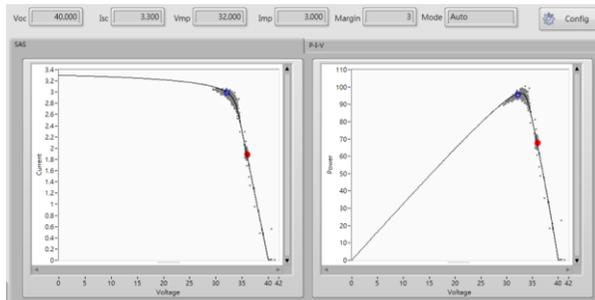


圖 8.11

MPPT 偏移模式

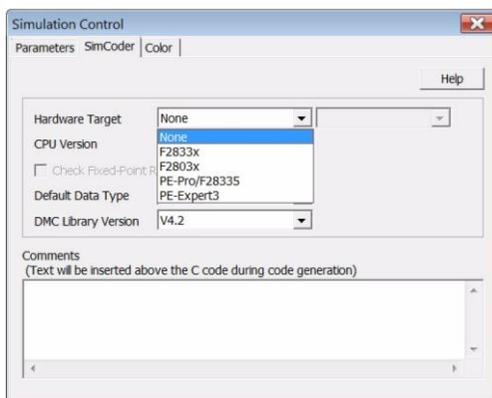


附錄 A SimCoder 概述

SimCoder 為 PSIM 軟體的一個附加工具模組。可由 PSIM 的電路圖產生 C 程式碼(code)，藉由特定的硬體標的資料庫，SimCoder 產生的 C 程式碼可直接在 DSP 硬體平台上執行。本手冊將說明如何使用 SimCoder。

SimCoder 於模擬控制上設定

SimCoder 在模擬控制方塊的 SimCoder 表格中完成參數設定，如下圖，須適當地設定參數以便能正確地產生程式碼。



設定過程說明如下

所支援的硬體標的(Hardware Targets)

- SimCoder 支援下表的硬體標的：
 - None
產生的程式碼只能用於模擬，無法用於任何特定硬體標的。
 - F2833x
產生的程式碼可用於 TI 的浮點 TMSF2833x 系列 DSP。
 - F2803x
產生的程式碼可用於 TI 的定點 TMSF2803x 系列 DSP。
 - PE-Pro/F28335
產生的程式碼可用於 PE-Pro/F28335 DSP 板，為一塊由 Myway 公司所生產的 DSP 板，它使用 TI 的浮點 DSP TMSF28335 和 Myway 的 PE-OS 資料庫。
 - PE-Expert3
產生的程式碼可用於 PE-Expert3 DSP 硬體，為一塊由 Myway 公司所生產的 DSP 開發平台，它使用 TI 的浮點 DSP TMS320C6713 和 Myway 的 PE-OS 資料庫。

專案組態
(Project Configuration)

針對 F2833x 與 F2803x 標的，專案組態可設定為 RAM Debug, RAM Release, Flash Release 或 Flash RAM Release。對於 PE-Expert3 標的，專案組態可設定為 PE-View9 或 PE-View8。

CPU 版本	<p>對於 F2803x 與 2833x，隨後的 CPU 版本可選擇。</p> <ul style="list-style-type: none">• 針對 F2833x，有 F28335、F28334 和 F28332 可供選擇。• 針對 F2803x，各種腳位配置的 F28030 到 F28035 可供選擇。
檢查定點範圍	<p>這只針對 F2803x 系列，如果選取此項目，SimCoder 在模擬時會檢查數據並提供數據範圍的列表。在此表中，接近或超過此範圍的資料會被特別標注。</p>
預設資料形式 (Default Data Type)	<p>當硬體標的為浮點形式，此部分會自動被選取。當沒有硬體標的或其為定點形式，則必須在下拉式選單中選擇一個可用的初始資料形式。</p> <ul style="list-style-type: none">• 針對 F2803x，有如下可供選擇：整數、IQ1、IQ2、...、IQ30。• 當沒有硬體標的，其選擇如下：浮動、整數、IQ1、IQ2、...、IQ30。
DMC 元件資料庫版本	<p>德州儀器的數位馬達控制(Digital Motor Control, DMC)資料庫是由 C 函數(或巨集)組成，針對 C2000 DSP 器件的馬達控制使用者所開發。為了善用這項資源，SimCoder 整合了 DMC 資料庫函數到 PSIM 的元件資料庫以便產生程式碼。</p> <p>TI 在某些巨集已經發佈不同的版本，SimCoder 支援 V4.0、V4.1、和 V4.2。</p>
 注意	<p>一旦選用某個版本，其他版本將會被關閉。</p>
注解	<p>注解區在 SimCoder 表格的下方可供使用者在 SimCoder 產生的程式碼中加入注解，此區的所有文字會被加入在 C 程式碼的開頭當作注解。</p>

產生程式碼的元件(Elements)

所有在 Elements>>Event Control 和 Elements>>SimCoder 中的元件皆可用於產生程式碼。

每個硬體標的元件可在 Elements>>SimCoder 的子選單 F2833x Target, F2803x Target, PE-Pro/F28335 Target 和 PE-Expert3 Target 中找到。

許多在 PSIM 標準資料庫中的元件也可用於產生程式碼，為了區別在標準資料庫中可用及不可用於產生程式碼的元件，在 Options>>Settings>>Advanced，如果選項框“*show image next to elements that can be used for code generation*”被選取，一個小圖案  會出現在這些可用於產生程式碼的元件前。

此外，當選取此選項框時，F2833x 和 F2803x Target 的所有元件前會出現  圖案，PE-Pro/F28335 Target 出現 ，PE-Expert3 Target 出現 。

在標準資料庫中可用於產生程式碼的元件列表，請參照 37 頁。

附錄 B 產生程式碼 - 逐步介紹

使用 SimCoder 自動產生程式碼的設計步驟如下：

1. 在 PSIM 下系統之控制部分使用連續模式進行設計與模擬。
2. 將系統的控制部分轉變為離散模式並重新模擬驗證。
3. 如無硬體標的亦可將控制部分置於子電路內並產生程式碼。
4. 如具有硬體標的，使用硬體標的元件去取代原系統之元件，並經由模擬驗證結果，最後產生程式碼。

然而，前兩步並非強制性，也可不經模擬直接建立線路圖並產生程式碼。



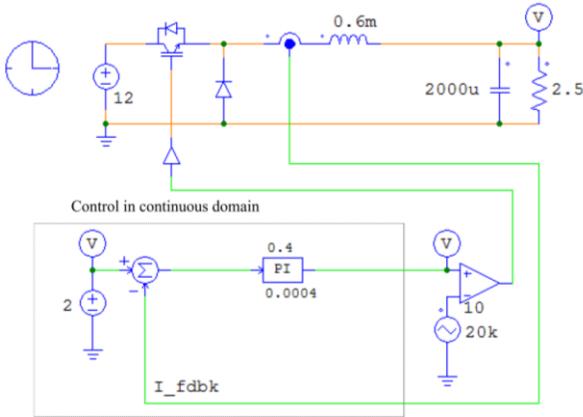
只有當控制部分在離散模式下才能產生程式碼，連續模式則不能。因此，對於 SimCoder 而言，Digital Control Module 是必備的。

下列章節將用簡單的例子說明產生程式碼的過程。

連續模式系統

通常系統會先在連續模式下設計與模擬，下面為一個簡單的電流回授直流轉換器電路，在連續模式下設計一個有 PI 控制器的控制電路，PI 的增益 k 為 0.4 與時間常數 T 為 0.0004，切換頻率為 20kHz。

此範例的目的為針對虛線框內的控制電路產生 C 程式碼，要產生程式碼，第一步為將 S 平面(S-domain)的類比 PI 控制器轉換成 Z 平面(Z-domain)的數位 PI 控制器。

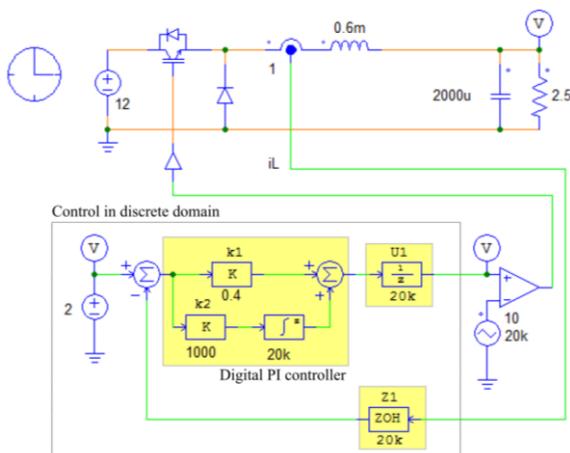


離散模式系統

要將類比控制器轉換成數位控制器，可使用 Digital Control Module 附帶的 *s2z Converter*，該程式可在 **Utilities>>s2z Converter** 啟動。

類比控制器可用不同的方法轉換成數位控制器，最常用的為 **Bilinear**(也可稱為 **Tustin** 或 **Trapezoidal**)法和 **Backward Euler** 法。

此範例中，取樣頻率與切換頻率同為 20kHz，使用 **Backward Euler** 法將類比 PI 控制器轉換成數位 PI 控制器。由轉換程式得知：PI 控制器的比例部分參數為 $k_1=0.4$ ，積分部分參數為 $k_2=1000$ ，數位 PI 控制器的電路如下圖



相較於連續模式下的控制電路，此電路有三處變更，以黃色框強調：

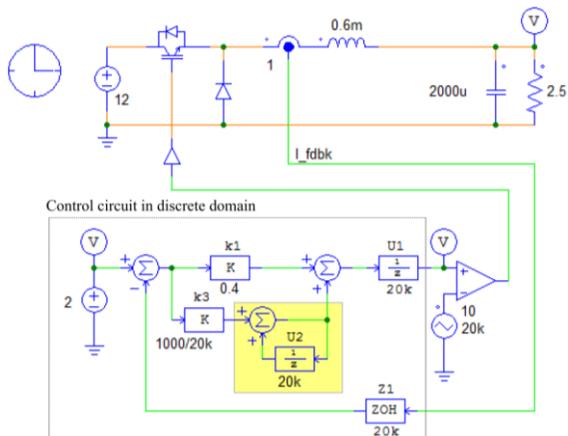
- (i) 數位 PI 控制器取代類比 PI 控制器，數位積分器的演算法標誌設定為 1(針對 **Backward Euler** 法)且取樣頻率設為 20kHz，由轉換程式取得的增益 k_1 和 k_2 敘述如上。
- (ii) 回授電流取樣 i_L 以 **zero-order-hold(ZOH)** 方塊 Z_1 來模擬在數位硬體實現的 A/D 轉換器。
- (iii) 數位控制內部一個週期延遲的實現以單位延遲(**unit-delay**)方塊 U_1 用來模擬。延遲的原因為，通常在一個週期開始時取樣，在週期內計算控制器參數，但因為完成計算需花時間，新的計算量通常等到下一個週期開始才使用。

請注意轉換後的數位控制器應產生穩定的控制迴路和所期望的性能，如果數位控制器模擬的結果不穩定或不如預期，則須回頭查看類比控制系統，重新設計類比控制器並重覆這個過程。

藉由 Backward Euler 法，也可在時域中將輸入輸出關係做如下描述：

$$y(n) = y(n-1) + T_s * u(n)$$

其中 $y(n)$ 和 $u(n)$ 為當時的輸出與輸入， $y(n-1)$ 為先前取樣週期的輸出， T_s 為取樣週期。使用上述的方程式，可用加法器(summer)和單位延遲(unity-delay)方塊取代上述電路的離散積分器，如下圖所示



注意

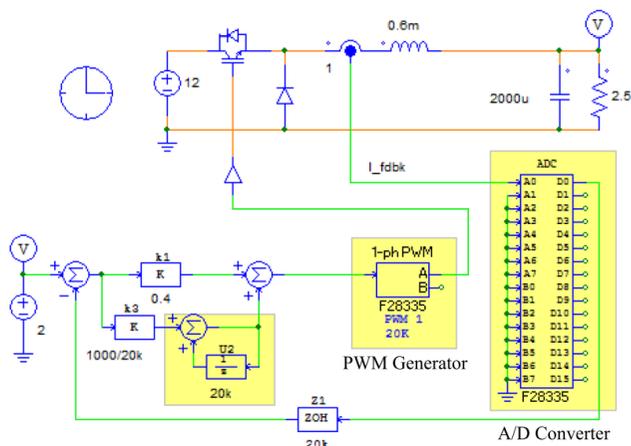
因為方程式中的 T_s 因子，比例方塊的增益 k_2 須除以 20kHz 的取樣頻率，此電路的優點在於更容易啟動或終止積分器積分。

有了離散模式下的控制電路，可往下進行下一個步驟。

產生硬體標的之程式碼

為產生特定硬體標的的程式碼，所設計電路須利用相關硬體標的的元件來修改，變數須因應硬體元件的有效範圍做適當的調整。

下面為同樣的範例電路但增加 F2833x 的硬體標的的元件，為了更清楚說明，硬體元件以黃色來標注。



此範例電路圖，做了如下變更：

1. 在電流感測器與控制子電路間加上 A/D 轉換器，須特別注意 A/D 轉換器的輸入範圍。如果電流感測器的輸出超過 A/D 轉換器的範圍，它必須做相對應的縮減，對於這個例子，A/D 轉換器的設定如下：
 - ADC Mode : Start-stop (8-channel)。這意味著 A/D 轉換由 PWM 產生器觸發，且只使用半數的 ADC 轉換器。
 - Ch A0 Mode : DC。設定輸入信號範圍為 0~+3V。
 - Ch A0 Gain : 1.0。

-
2. 以硬體 PWM 產生器取代比較器與載波，此範例的硬體 PWM 產生器相關設定如下：
 - PWM Source : PWM1。定義 F28335 處理器的 PWM 模組。
 - Output Mode : Use PWM A。定義 PWM 的輸出端口。
 - Sampling Frequency : 20k。定義取樣頻率為 20kHz。
 - Carrier Wave Type : Sawtooth(start high)。此設定往後在附錄 E F2833x 硬體標的會說明。
 - Trigger ADC : Trigger ADC Group A。此設定往後在附錄 E F2833x 硬體標的會說明。
 - ADC Trigger Position : 0。此設定往後在附錄 E F2833x 硬體標的會說明。
 - Peak to Peak Value : 10。定義 PWM 產生器的鋸齒波信號範圍。
-
3. 因為 PWM 產生器本身就包含一個取樣週期的延遲，故移除前一圖中之單位延遲方塊 U1。
-
4. 在 Simulation Control 中的 SimCoder 選項表：
 - Hardware Type 設定 F2833x 以及 RAM Debug。
 - CPU Version 設定 F28335。
 - Default Fixed-Point Position 並不適用，因為它是浮點。
-

5. 使用者可在產生的程式碼前加上註解 (comment)，進入 Simulation Control 的 SimCoder 選項表，便可輸入或編輯註解。

為了在增加硬體元件後檢查這些變更是否有效，執行全系統模擬，其結果應該很接近 2.2 節數位控制系統所模擬的結果。

一旦驗證了模擬結果，可點選 **Simulate>> Generate Code** 產生 C 程式碼，針對 F2833x 硬體所產生的程式碼可不須做任何變更便可直接燒錄。

對於上述系統，SimCoder 產生的程式碼如下

```
/*
// This code is created by SimCoder Version 9.3.3 for TI F2833x Hardware Target
//
// SimCoder is copyright by Powersim Inc., 2009-2013
//
// Date: February 24, 2014 14:36:33
*/
#include<math.h>
#include"PS_bios.h"
typedef float DefaultType;
#defineGetCurTime() PS_GetSysTimer()

interrupt void Task();

DefaultTypefGbliref = 0;
DefaultTypefGblU2 = 0;

interrupt void Task()
{
    DefaultTypefU2, fSUMP1, fSUMP3, fk3, fk1, fSUM1, fZ1, fTI_ADC1, fVDC2;

    PS_EnableIntr();
    fU2 = fGblU2;
}
```

```
FTI_ADC1 = PS_GetDcAdc(0);
fVDC2 = 2;
fZ1 = fTI_ADC1;
fSUM1 = fVDC2 - fZ1;
fk1 = fSUM1 * 0.4;
fk3 = fSUM1 * (1000.0/20000);
fSUMP3 = fk3 + fU2;
fSUMP1 = fk1 + fSUMP3;
PS_SetPwmRateSH(fSUMP1);
#ifdef DEBUG
fgbliref = fVDC2;
#endif
#endif
fgblU2 = fSUMP3;
PS_ExitPwmGeneral();
}

void Initialize(void)
{
    PS_SysInit(30, 10);
    PS_StartStopPwmClock(0);
    PS_InitTimer(0, 0xffffffff);
    PS_InitPwm(1, 0, 20000*1, (4e-6)*1e6, PWM_POSI_ONLY, 42822);// pwnNo, waveType, frequency, deadtime,
outtype
    PS_SetPwmPeakOffset(1, 10, 0, 1.0/10);
    PS_SetPwmIntrType(1, ePwmIntrAdc0, 1, 0);
    PS_SetPwmVector(1, ePwmIntrAdc0, Task);
    PS_SetPwmTzAct(1, eTZHighImpedance);
    PS_ResetAdcConvSeq();
    PS_SetAdcConvSeq(eAdc0Intr, 0, 1.0);
    PS_AdInit(1, 1);

    PS_StartStopPwmClock(1);
}

void main()
{
    Initialize();
    PS_EnableIntr(); // Enable Global interrupt INTM
    PS_EnableDbgm();
    for (;;) {
    }
}
```

產生的程式碼有著如下的結構：

- *Interrupt void Task()*：中斷服務程序為 20kHz，每 20kHz 被呼叫一次。
- *void Initialize()*：初始化程序，它將對硬體初始化。
- *void main()*：主程式，他呼叫初始化程序，並執行一個無限迴圈。



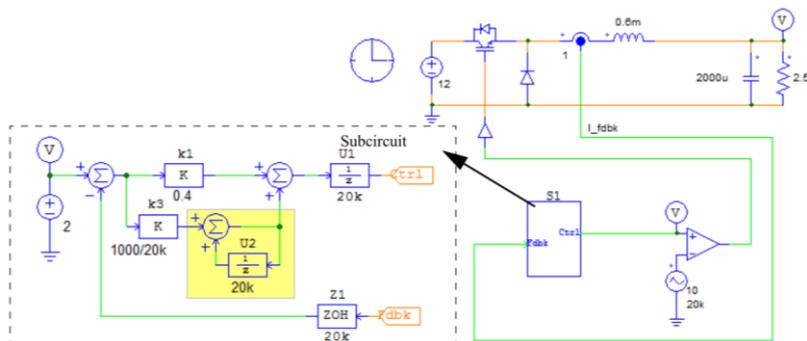
在此範例中，所有的控制方塊都在 20kHz 的取樣頻率下運作，如果有方塊在不同的取樣頻率下運作，將會建立另一個服務程序。一個中斷服務程序對應一個取樣頻率。對於那些沒有相對應取樣頻率的方塊，其對應的程式碼會被放在主程式內。

這個程式碼和所有必須的文件儲存在與原電路相同目錄的子資料夾下，使用者可在 TI Code Composer Studio(CCS)環境中載入這些文件，編譯程式碼，並燒錄到 DSP 硬體做即時操作。

子電路(Subcircuit)產生程式碼

在 PSIM 中，可以將電路的某部分做成一個子電路，SimCoder 可以針對每個子電路產生程式碼，前提是子電路中的每個元件都可支援產生程式碼，在 2.5.3 節會列出一些限制。

為了說明子電路產生程式碼，將持續使用 2.3 節的範例電路，虛線框內的控制電路將會置換成一個子電路，如下圖



選取虛線框內的電路以產生子電路。擊點滑鼠右鍵將會出現一個下拉式選單，選擇 **Create Subcircuit** 並設定子電路名稱。

子電路產生程式碼不包含比較器和 PWM 載波源。這樣安排的一個原因是，大部分的硬體設置這兩個功能由外部硬體實現或嵌入至微控制器的周邊介面。另一個原因是，程式碼每 20kHz 的取樣頻率才執行一次，但載波和比較器在每個模擬時間間隔(time step)都需計算。當使用 SimCoder 產生程式碼，每個元件的取樣頻率都需要定義。比較器的兩個輸入，一個是 20kHz 取樣頻率的控制器，另一個是未定義的載波源。在這種情形下，SimCoder 會假定比較器的兩個輸入都有相同的取樣頻率，由有定義取樣頻率者決定。

SimCoder 可針對子電路產生程式碼做為模擬或具有硬體標時使用，這兩種形式的程式碼不可交替使用。針對模擬產生的程式碼不可用在硬體標的，反之亦然。這兩種情形再下面章節或說明。

子電路為模擬產生程式碼

針對模擬產生程式碼，步驟如下

1. 在主電路中，在子電路方塊擊點滑鼠右鍵並選擇 **Attributes**。
 2. 在 **Attribute** 對話視窗下，擊點按鈕 **Generate Code**，並選擇 **Generate Code for Simulation**。
 3. 如果需要，使用者可在程式碼的開頭增加註解。在 **Simulation Control** 的 **SimCoder** 表，擊點 **Generate Code** 按鈕之前在對話視窗輸入或編輯註解。
-

產生的程式碼開頭如下：

```

/*****
// This code is created by SimCoder Version 9.3.3
//
// SimCoder is copyright by Powersim Inc., 2009-2013
//
// Date: February 24, 2014 14:55:33
*****/
#include <stdio.h>
#include <math.h>
#define ANALOG_DIGIT_MID 0.5
#define INT_START_SAMPLING_RATE 1999999000L
#define NORM_START_SAMPLING_RATE 2000000000L

typedef void (*TimerIntFunc)(void);
typedef double DefaultType;
DefaultType *inAry = NULL, *outAry;
DefaultType *inTmErr = NULL, *outTmErr;

double fCurTime;
double GetCurTime() {return fCurTime;}

/* The input/output node definition for C/DLL block.
   The 2nd display node (outAry[1]): From element 'S1_iref'.
*/
/* The C block for the generated code has the following additional output port(s):
   2 - S1.iref
*/
void _SetVP6(int bRoutine, DefaultType fVal);
void InitInOutArray()
{... ..}
void FreeInOutArray()
{ ... ..}

void CopyInArray(double* in)
{ ... ..}

void CopyOutArray(double* out)
{ ... ..}

void Task();
void TaskS1(DefaultType fIn0, DefaultType *fOut0);

DefaultType fGblS1_U1 = 0;
DefaultType fGblS1_U2 = 0;

void TaskS1(DefaultType fIn0, DefaultType *fOut0)
{ ... ..}

void Task()
{
    TaskS1(inAry[0], &outAry[0]);
}

typedef struct {
    TimerIntFunc func;
    long samprate;
    double tmLastIntr;
} TimeChk;
#define NUM_TIMER_INTR 1
TimeChk lGbl_TimeOverChk[NUM_TIMER_INTR] = {
    {Task, 20000, 0}};

void InitAllTaskPtr(void)
{
    lGbl_TimeOverChk[0].func = Task;

```

```
        val = fVal;
    }
    outAry[1] = val;
}
```

由子電路產生的程式碼結尾，包含 **SimulationStep** 函數、**SimulationBegin** 函數以及 **SimulationEnd** 函數，如下所示。這些函數可被用在能取代子電路的 C Block 中。

```
void SimulationStep(
    double t, double delt, double *in, double *out,
    int *pnError, char * szErrorMsg,
    void ** reserved_UserData, int reserved_ThreadIndex, void * reserved_AppPtr)
{ ... ...}

void SimulationBegin(
    const char *szId, int nInputCount, int nOutputCount,
    int nParameterCount, const char ** pszParameters,
    int *pnError, char * szErrorMsg,
    void ** reserved_UserData, int reserved_ThreadIndex, void * reserved_AppPtr)
{
    InitInOutArray();
}

void SimulationEnd(const char *szId, void ** reserved_UserData, int reserved_ThreadIndex, void *
reserved_AppPtr)
{
    FreeInOutArray();
}
```

以 C Block 取代子電路

在子電路屬性的對話視窗，一個叫 **Replace subcircuit with generated code for simulation** 的選擇框。如選取時，模擬完成時系統將以 C Block 取代子電路。使用者在系統中不需要經歷以 C Block 取代子電路的過程。

然而，如果使用者想隨意地修改 C 程式碼，可經由下列步驟將產生的程式碼放到 C Block。

在上面的例子，由子電路產生的程式碼包含四個部分：**SimulationStep**、**SimulationBegin**、**SimulationEnd** 以及餘下的部分。同樣地，C Block 也由這四部分組成，包含下列部分：

```

#include <Stdlib.h>
#include <String.h>
#include <math.h>
#include <Psim.h>

// PLACE GLOBAL VARIABLES OR USER FUNCTIONS HERE...
... ..

////////////////////////////////////
// FUNCTION: SimulationStep
{
// ENTER YOUR CODE HERE...

}

////////////////////////////////////
// FUNCTION: SimulationBegin
{
// ENTER INITIALIZATION CODE HERE...

}

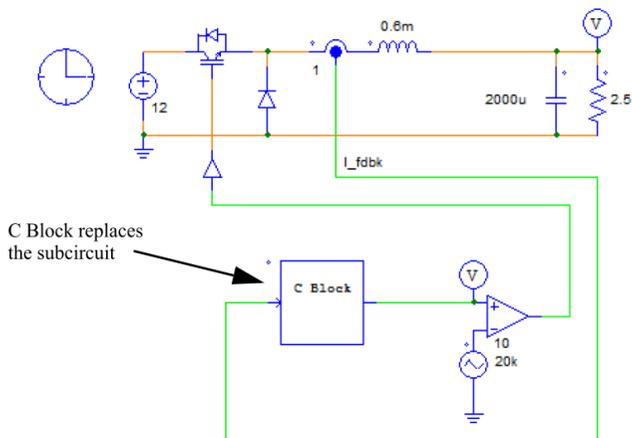
////////////////////////////////////
// FUNCTION: SimulationEnd
{

}

```

在主電路上建立一個 C Block，使用下拉式選單至 **Element>>Other>> Function Blocks>>C Block**，複製所產生程式碼的每一部分到 C Block，從 *SimulationStep* 到 C Block 的 *SimulationStep*，從 *SimulationBegin* 到 C Block 的 *SimulationBegin*，從 *SimulationEnd* 到 C Block 的 *SimulationEnd*，再將餘下的程式碼放到 *User Functions* 部份。

以 C Block 取代子電路的範例電路如下：



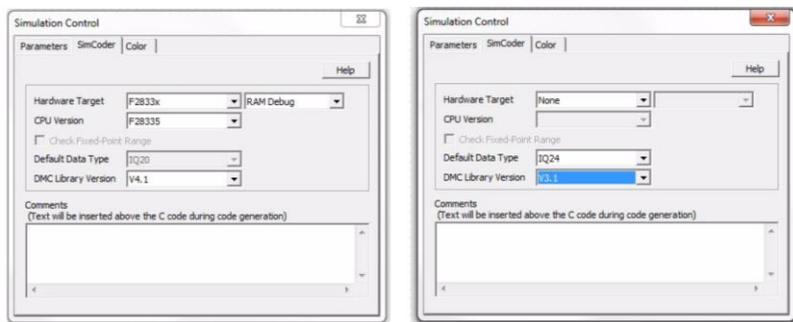
子電路為硬體標的產生程式碼

與 2.5.1 節同樣的範例電路也可用來為硬體電路產生程式碼。

在 Simulation Control 對話視窗的 SimCoder 表內設定所需的硬體標的類型，如下圖左，選定 F2833x，CPU 版本為 F28335，專案組態為 RAM Debug。

當硬體標的選擇 None，仍可針對特定的預設資料格式產生硬體標的程式碼。

下圖右的範例以 IQ24 的位置設定定點的資料格式。



開啟子電路屬性視窗，擊點 *Generate Code* 並選擇 *Generate Code for Hardware Target*。

SimCoder 用子電路針對 F28335 所產生的 C 程式碼如下，不同於針對整個系統所產生的程式碼，子電路產生的程式碼沒有主程式和初始化程序。它可插入使用者自己的 F28335 硬體標的程式碼。

子電路只有一個取樣頻率，正因如此，產生的程式碼只有一個 *TaskS1* 函數，變數 *fnIn0* 表示為子電路輸入 *I_fdbk*，變數 *fOut0* 則對應子電路輸出 *Ctrl*。

```
#define NULL (0)
#ifndef DSP28_DATA_TYPES
#define DSP28_DATA_TYPES
typedef int          int16;
typedef long        int32;
typedef long long   int64;
typedef unsigned int  Uint16;
typedef unsigned long  Uint32;
typedef unsigned long long  Uint64;
typedef float         float32;
typedef long double   float64;
#endif
DefaultTypefGblS1_U1 = 0;
DefaultTypefGblS1_iref = 0;

// Parameter list for S1
void TaskS1(DefaultType fIn0, DefaultType *fOut0)
{
    DefaultTypefS1_U1, fS1_SUMP1, fS1_B4, fS1_k2, fS1_k1, fS1_SUM1, fS1_Z1;
    DefaultTypefS1_VDC2;

    *fOut0 = fGblS1_U1;

    fS1_VDC2 = 2;
    fS1_Z1 = fIn0;
    fS1_SUM1 = fS1_VDC2 - fS1_Z1;
    fS1_k1 = fS1_SUM1 * 0.4;
    fS1_k2 = fS1_SUM1 * 1000;
    {
        static DefaultType out_A = 0;
        fS1_B4 = out_A + 1.0/20000 * (fS1_k2);
        out_A = fS1_B4;
    }
    fS1_SUMP1 = fS1_k1 + fS1_B4;
    fGblS1_U1 = fS1_SUMP1;
#ifdef _DEBUG
    fGblS1_iref = fS1_VDC2;
#endif
}
```

子電路產生程式碼的限制

使用 SimCoder 子電路產生程式碼有幾個限制，條列如下：

- 所有在子系統的元件都需支援產生程式碼，要得知元件是否可用於產生程式碼，在 PSIM 中，至 **Options>>Settings**，選取選擇框 **Show image next to elements that can be used for code generation**。在 PSIM 元件庫中所有前端有標式符號的元件都可用來產生程式碼。
 - 只可使用單方向的子電路端口，也就是說，輸入信號端口只可用於子電路輸入，輸出信號端口只可用於子電路輸出，雙向端口不可使用。
 - 子系統中不可放置輸入/輸出元件(如 A/D 轉換器，數位輸入/輸出，編碼器，計數器和 PWM 產生器)和硬體中斷元件，這些都只能在最上層的主電路中。
 - 如果子系統的輸入有一取樣率，且無法由子系統內部電路產生，則必須在輸入端連接一個 **zero-order-hold** 方塊以明確定義取樣率。如果這個範例不使用 **zero-order-hold** 方塊，輸入(以及之後連接到此輸入的方塊)將被視為沒有取樣率。
 - 如果子系統的輸入沒有 **zero-order-hold** 方塊與之連接，**SimCoder** 將會在子系統中由連接到它的方塊獲得取樣率，然而，避免混淆，強烈建議在每一個輸入放置 **zero-order-hold** 方塊以明確定義取樣率。
-

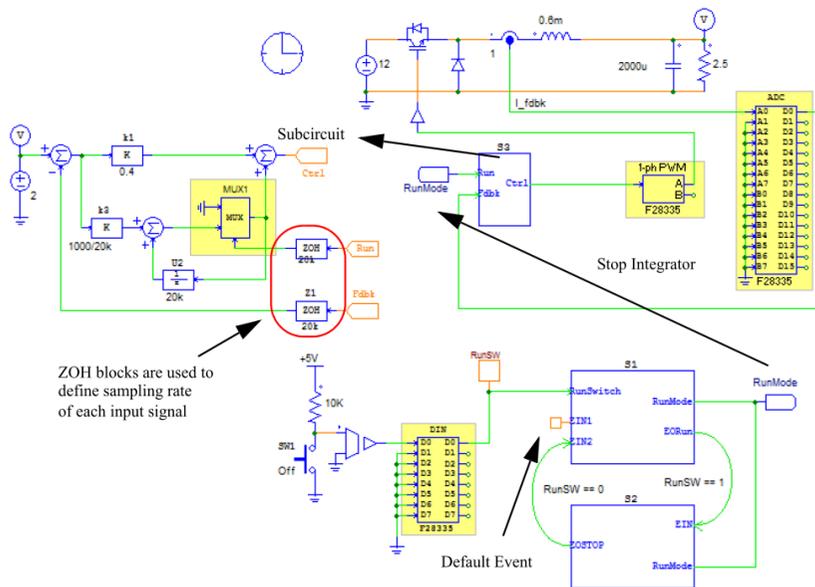
事件控制(Event Control)的系統

通常一個系統會包含事件轉換，當某些條件成立時，系統會從這個狀態轉換至另一個狀態，SimCoder 經由子電路處理事件控制，事件控制會在附錄 C 事件處理做更詳細地說明。

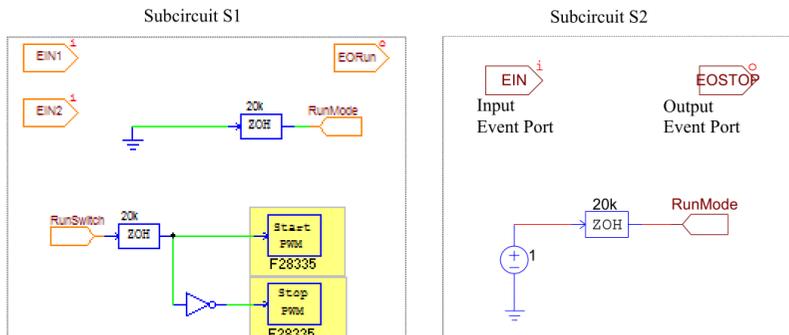
在這節的範例增加下列的情況以便說明事件控制如何執行：

- 系統的開始與停止由一個手動開關控制，因此，此系統會有兩個操作模式：停止模式與執行模式。當開關切換時，系統將由一個模式轉換到另一個模式。
- 在停止模式，為了避免積分器飽和，輸出將會重置為 0。事件控制系統範例如下：

事件控制系統範例如下：



如下圖表，S1 和 S2 為子電路，其內容如下：



相較於電路部分，改變如下：

- 增加兩個事件控制子電路以實現兩種操作模式：停止模式(子電路 S1)與執行模式(子電路 S2)。
- 預設的操作模式為停止模式。這是經由連接 Default Event 元件到子電路 S1 的端口 EIN1 來定義。
- 子電路 S1 有兩個輸入事件端口為 EIN1 和 EIN2，一個輸出事件端口 EORun，一個輸入信號端口 RunSwitch，以及一個輸出信號端口 RunMode。子電路 S2 有一個輸入事件端口 EIN，一個輸出事件端口 EOSTOP，以及一個訊號端口 RunMode。
- 定義從停止模式轉換到執行模式的條件，反之亦然。變數 RunSW 在這種條件下為一全域變數(更詳盡請參照 72 頁)，且藉由全域變數元件連接到數位輸入元件的輸出腳位 D8 來定義。
- 硬體數位輸入元件用來量測按鈕開關 SW1 的位置，當開關截止，數位輸入電壓為高電位(1)，因此為全域變數 RunSW 與系統處於執行模式。當 RunSW 為低電位(0)，系統處於停止模式。

- 在停止模式下增加多工器 MUX1 以避免積分器積分，RunMode 信號為 0 且積分器不動作當系統不執行時。當 RunMode 信號為 1，積分器將開始動作。

系統如何動作如下：

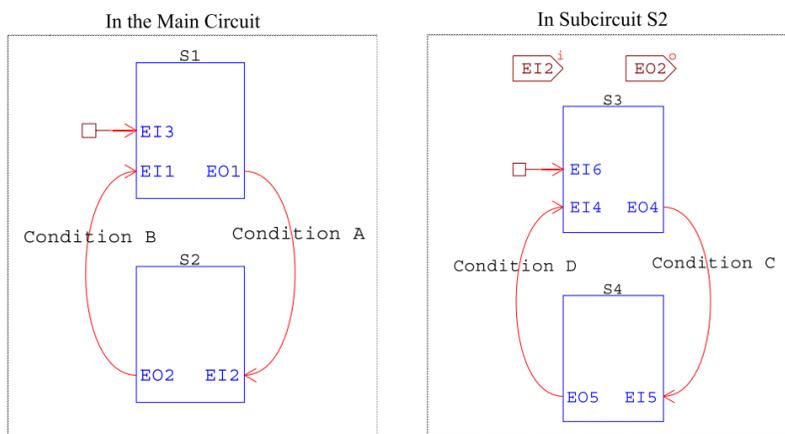
- 手動開關的位置經由硬體數位輸入讀取，信號經由輸入信號端口 RunSwitch 傳送給子電路 S1(停止模式)，同樣的訊號也被指定為全域變數 RunSw。
- 系統初始在停止模式，當條件 RunSW==1(或 RunSW 為 1)成立，系統將從停止模式轉換成執行模式，由 S1 的輸出事件端口 EORun 與 S2 的輸入事件端口 EIN 的連接來定義。
- 當處於執行模式，如 RunSW==0(或 RunSw 為 0)的條件成立，系統將會從執行模式轉換成停止模式，由 S2 的輸出事件端口 EOSTOP 與 S1 的輸入事件端口 EIN2 的連接來定義。
- 在停止模式的子電路，RunMode 訊號會被設定為 0，只要 RunSwitch 的信號為 0，硬體 PWM 產生器將會停止，但當 RunSwitch 變為 1 時，PWM 開始動作，同時由停止模式切換到執行模式。
- 在執行模式的子電路，為了能使積分器動作，RunMode 信號將設定為 1。

在系統修正後，使用者可在針對硬體標的產生系統程式碼前藉由模擬來驗證變更是否正確。

附錄 C 事件處理

基本概念

事件用來描述系統由一個狀態轉變到另一個狀態。舉例來說，下圖顯示幾個操作狀態以及轉換如何發生。



在主電路有兩個狀態 S1 和 S2，都是子電路形式。每個狀態的原理圖都包含一個子電路。狀態 S1 有兩個輸入事件端口(port) EI1 和 EI3，一個輸出事件端口 EO1；狀態 S2 有一個輸入事件端口 EI2，一個輸出事件端口 EO2。一開始的預設狀態為 S1，其乃由預設事件元件連接到輸入事件端口 EI3 所定義。

藉由轉換條件 A，S1 的輸出事件端口 EO1 連接到 S2 的輸入事件端口 EI2，這意味著當條件 A 成立時，系統將從狀態 S1 轉換成狀態 S2。同樣地，S2 的輸出事件端口 EO2 連接到輸入事件端口 EI1，當條件 B 成立時，系統將由狀態 S2 轉換成 S1。

當兩個或多個狀態不能同時存在且在任何時間點只存在一個狀態，如這個例子的 S1 和 S2，稱這些狀態為專屬狀態。

右邊的系統顯示子電路 S2 內部，依序有兩個狀態 S3 和 S4，當系統轉換到 S2 時，預設由 S3 開始。當條件 C 成立時，將由 S3 轉換到 S4，當條件 D 成立時，又再回到狀態 S3。

一個系統包含的狀態數量沒有限制。

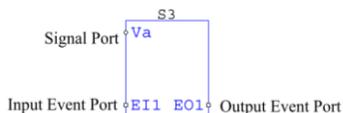
事件處理的元件

下列元件可用來定義事件和狀態轉換：

1. 輸入事件端口(Input event port)
2. 輸出事件端口(Output event port)
3. 初始事件元件(Default event element)
4. 方塊初次進入的旗標(Flag for block first entry)
5. 硬體中斷元件(Hardware interrupt element)
6. 全域變數(Global variable)

在子電路內放置輸入事件端口將創造一個允許轉入子電路的事件，同樣地，在子電路內放置輸出事件端口將創造一個允許轉出子電路的事件。

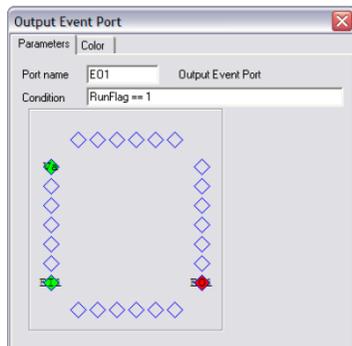
舉例來說，下圖為放置輸入事件端口和輸出事件端口後的子電路圖示。



事件端口的圖示為方形，不同於圓形的信號端口圖示。

連接到輸入事件端口的只能是輸出事件端口或硬體中斷元件並使用事件連接線功能。輸入/輸出事件端口和硬體中斷元件不可連接到其他形式的接點上。

必須定義每個輸出事件端口的條件。子電路 S3 的輸出事件端口 *EO1* 屬性視窗如下



“RunFlag==1”為觸發輸出事件發生的條件，條件敘述必須為有效的 C 程式碼表示式。舉例來說，條件敘述可以如下：

`(RunFlag==1)&&(Flag>=250.) || (FlagB<Vconst)`



注意

只有在參數文件中定義或通過主電路進入子電路的全域變數、數值和參數常數可使用在條件表示式。在上面的表示式，RunFlag、FlagA 和 FlagB 為全域變數，Vconst 為參數文件中定義或通過主電路進入子電路的常數。

連接全域變數元件到一個節點可建立一個全域變數。

具有事件的子電路之限制

包括輸入或輸出事件端口的子電路會被視為具有事件的子電路，具有事件子電路的內部所有元件將會繼承事件屬性。也就是說，如果子電路 A 在子電路 B 內，子電路 B 是具有事件的子電路，即使子電路 A 沒有任何輸入/輸出端口，仍被視為具有事件的子電路。

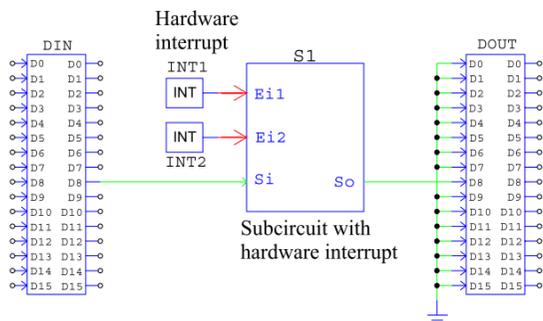
當用子電路來處理事件，在 PSIM 中有三種類型的子電路：

- 正規子電路(*Regular subcircuits*)：這種類型的子電路不包含任何事件端口且與先前常見的子電路相同。
- 具有事件的子電路(*Subcircuit with events*)：這種類型的子電路包含輸入/輸出事件端口，但輸入事件端口並沒有連接硬體中斷元件。
- 具有硬體中斷的子電路(*Subcircuit with hardware interrupt*)：這種類型的子電路只包含輸入事件端口並只連接到硬體中斷元件，子電路內無輸出事件端口，且輸入事件端口沒有連接到輸出事件端口。這是具有事件的子電路中的特例，這種類型的子電路指專門處理硬體中斷。

因為具有事件或硬體中斷的子電路只包含在產生程式碼中，針對這兩種類型的子電路，更多的限制如下：

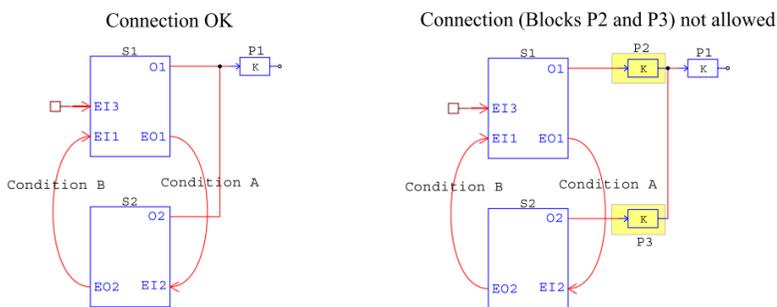
- 具有事件或硬體中斷的子電路內的所有元件必須支援產生程式碼。例如，這樣的子電路不能包括電阻和 RMS 方塊，因為這兩個都不支援產生程式碼。
- 具有硬體中斷的子電路可有多個輸入事件端口，但不能有輸出事件端口。也只有硬體中斷元件可以連接到輸入事件端口。此外，子電路的信號輸出/輸入端口只可連接到硬體元件，不可連到其他函數方塊。

下圖顯示一個具有硬體中斷的子電路如何連接：



具有硬體中斷的子電路 S1 有兩個硬體中斷元件 INT1 和 INT2 與之連接，一個信號輸入端口 Si 連接到硬體數位輸入，一個信號輸出端口 So 連接到硬體數位輸出。

- 如果具有硬體中斷的子電路包含有取樣率的 Z 平面方塊，當硬體中斷發生時，子電路被呼叫取樣率將被忽略。舉例來說，如果一個包含數位積分器的子電路，數位積分器的取樣率將會被忽略。在積分器計算時，中斷前的值將被保留成硬體中斷後的值。
- 如果有兩個子電路的輸出信號相連接，則必需直接連接，不可經由其他元件連接，參考下圖說明：



在左邊的電路，子電路 S1 和 S2 都有一個信號輸出端口 O1 和 O2，它們一起連接到比例方塊 P1 的輸入，這種電路動作的方式為方塊 P1 的輸入來自 O1 或 O2 端口，取決於哪個狀態動作，這種連接方式是允許的。

然而，在右邊的電路，O1 連接到 P2，O2 連接到 P3，P2 和 P3 的輸出連接在一起，這樣的連接是不被允許的。在此例子中，方塊 P2 應該移到子電路 S1 內，方塊 P3 則應該移動到子電路 S2 內。

附錄 D SimCoder 資料庫

SimCoder 有無硬體標的皆可使用，當沒有硬體標的時，將控制原理圖轉成 C 程式碼，此程式碼在 PSIM 中模擬時，並不針對特定硬體，另一方面，當有硬體標的時，SimCoder 會產生已準備好在特定硬體標的上執行或被特定硬體所採用的程式碼。

SimCoder 元件資料庫包含兩種類型的元件：一種為獨立於任何硬體標的或所有硬體標的皆可共用，另一種為針對特定的某個硬體標使用。

獨立於任何硬體標的之 SimCoder 元件包含如下：

- 某些 PSIM 標準資料庫的元件。
- 在 Elements>>Event Control 下的所有元件。
- 在 Elements>>SimCoder 下的全域變數(*Global Variable*)元件。

所有硬體標的皆可共用的 SimCoder 元件包含如下：

- 在 Elements>>SimCoder 下的中斷(*Interrupt*)元件。
 - 在 Elements>>SimCoder>>TI DMC Library 下的 *TI DMC* 元件。
-

針對特定硬體使用的 SimCoder 元件包含如下：

- F2833x Target：在 Elements>>SimCoder>>F2833x Target 下的所有元件。
 - F2803x Target：在 Elements>>SimCoder>>F2803x Target 下的所有元件。
 - PE-Pro/F2833x Target：在 Elements>>SimCoder>> PE-Pro/F2833x Target 下的所有元件。
 - PE-Expert3 Target：在 Elements>>SimCoder>> PE-Expert3 Target 下的所有元件。
-

所有獨立於硬體標的或硬體標的共用的 SimCoder 元件將在本章節說明，每個特定硬體標的之元件將在 5~8 章說明，TI DMC 資料庫則在第 [附錄 D](#) 說明。

標準 PSIM 資料庫的元件

許多在 PSIM 標準資料庫的元件都可用來產生程式碼，在 **Options>> Settings>> Advanced**，如果選項框“*show image next to elements that can be used for code generation*”被選取，一個小圖案會出現在這些可用於產生程式碼的元件前。



注意

所有的數學函數方塊和簡易 C 方塊，變數 t (針對時間)， Δ (針對時間步長) 不能用於 SimCoder 產生程式碼。

此外，參數文件元件和鋸齒電壓源元件在 SimCoder 中有特別的用法，將在這章節中說明。

在參數文件(Parameter File)中定義全域(Global)參數

參數文件元件可如同先前的用法，在 SimCoder，可用來定義全域參數。

為了使產生的程式碼更具可讀性與易於管理，有時候最好是在程式碼中以參數名稱取代實際數值。例如，如果控制器的增益是 1.23，定義參數 $K_p=1.23$ ，在程式碼中以參數 K_p 取代。

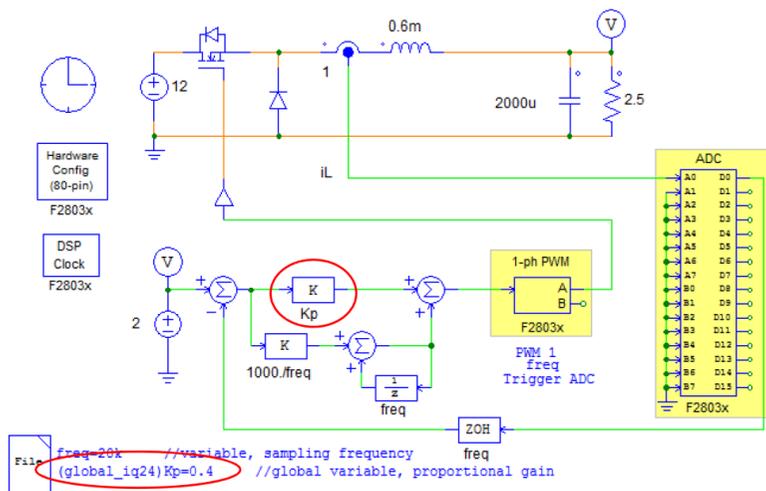
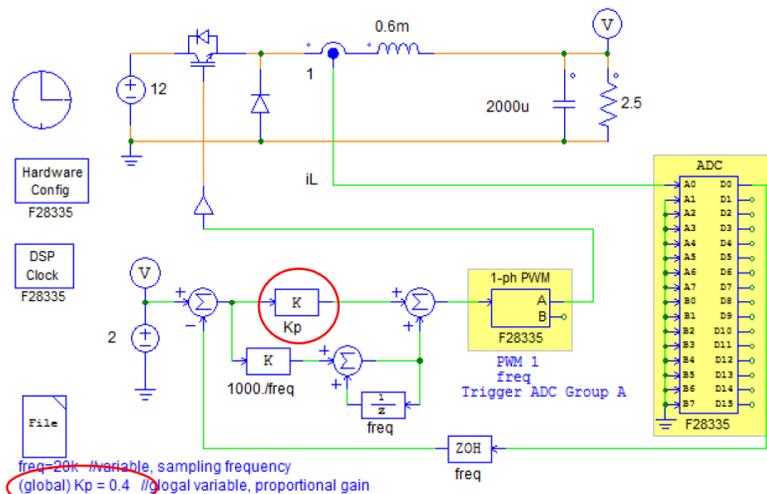
這種參數類型為全域的，可在程式碼中的任何地方使用。定義一個參數為全域參數，在參數文件內容中，參數名稱前使用“(global)”定義。

如下所示為一降壓式轉換器電路，比例控制器的增益定義為 K_p ，在參數文件中，如果 DSP 標的為浮點 F28335，參數 K_p 定義為：

$$(\text{global})K_p=0.4$$

如果 DSP 標的為在 IQ-24 的定點 F28035，參數 K_p 定義為：

$$(\text{global_Iq24})K_p=0.4$$



注意

在程式碼中，參數 K_p 的初始值定義為 0.4，參數名稱 K_p 則用在計算中。

針對 F28335 產生的程式碼如下

```
/**
 * This code is created by SimCoder Version 9.1 for TI F28335 Hardware Target
 * SimCoder is copyright by Powersim Inc., 2009-2011
 * Date: November 08, 2011 11:26:37
 */
#include <math.h>
#include "PS_bios.h"
typedef float DefaultType;
#define GetCurTime() PS_GetSysTimer()

interrupt void Task();

DefaultType fGbliref = 0.0;
DefaultType fGblUDELAY1 = 0;

DefaultType Kp = 0.4; // The global parameter Kp is defined here.
interrupt void Task()
{
    DefaultType fVDC2, fTI_ADC1, fZOH3, fSUM1, fP2, fSUMP3, fUDELAY1, fP1, fSUMP1;
    PS_EnableIntr();
    fUDELAY1 = fGblUDELAY1;

    fTI_ADC1 = PS_GetDcAdc(0);
    fVDC2 = 2;
#ifdef _DEBUG
    fGbliref = fVDC2;
#endif

    fZOH3 = fTI_ADC1;
    fSUM1 = fVDC2 - fZOH3;
    fP2 = fSUM1 * (1000./20000);
    fSUMP3 = fP2 + fUDELAY1;
    fGblUDELAY1 = fSUMP3;
    fP1 = fSUM1 * Kp; // The parameter Kp is used here.
    fSUMP1 = fP1 + fSUMP3;
    PS_SetPwm1RateSH(fSUMP1);
    PS_ExitPwm1General();
}
```

產生鋸齒波

鋸齒波在硬體使用中可當做系統的計時器，或產生其他的週期波(如正弦波)，在 **Elements>>Sources>>Voltage** 下的鋸齒電壓源可在硬體中以實際的計數器實現。

假設在硬體中存在一個 32 位元的計數器，每 20ns 增加一次以產生鋸齒波。對於 PE-Expert3 硬體標的，在 PEV 板使用 32-bit 的非同步計數器，每 20ns 增加一次以產生鋸齒波。

事件控制元件

下列元件用來實現事件控制：

輸入事件

輸入事件元件圖示如下：

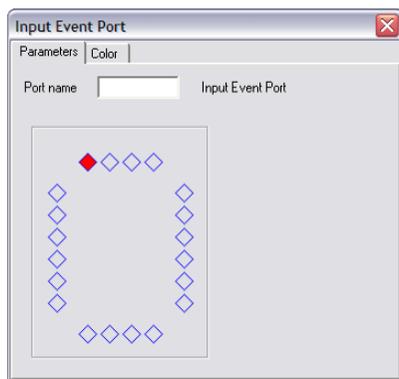
圖示

Input Event



圖示中的字母“i”代表“input”。

輸入事件元件為子電路界面端口的一種類型，只用在子電路內，連續擊點元件兩下後，即可定義端口的名稱及位置，如下：



在主電路中呼叫子電路，如果有事件連接線連接到這個端口，當事件連接的條件成立，系統將會經由輸入事件端口轉換到子電路。

輸出事件

輸出事件元件圖示如下：

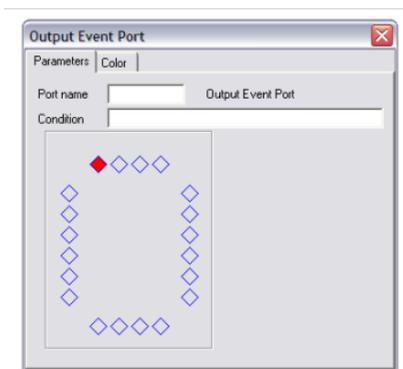
圖示

Output Event



圖示中的字母“o”代表“output”。

輸出事件元件亦為子電路界面端口的一種類型，只用在子電路內，連續擊點元件兩下後，即可定義端口的名稱及位置，如下：



當輸出事件端口定義的條件成立，系統將由這個子電路轉出至另一子電路。

條件敘述必須使用 C 語言所支援的格式及運算。例如，下面的敘述為可接受的條件敘述：

$$A==1$$

$$B>=1$$

$$(A>B)\&\&(C>D)$$

其中 A、B、C 和 D 為全域變數或常數。

初始事件

初始事件元件圖示如下：

圖示：

圖示

Default Event



當有多個專屬狀態時，初始事件元件用來定義哪個狀態為初始狀態，連接到這個子電路以外的子電路之輸入事件端口。

事件連結

事件連結元件為 SimCoder 中連接輸出事件端口或硬體中斷元件到輸入事件端口的元件。注意不要與一般用來連接其他 PSIM 元件的佈線工具混淆，事件連接只可用來連接事件。事件連結元件位於 **Elements>>Event Control>>Event Connection**。

在事件連接線連續擊點兩下，可編輯事件線所連接到的輸出事件端口之條件敘述。

除了起點和終點，事件連接線在此之間還有兩個點。藉由修正這兩點的位置來改變連接線的形狀，修改這兩點來以凸顯事件接線。擊點右鍵並選擇“Modify handle 1”或“Modify handle 2”。

事件方塊首次進入的標誌

有時候當程式第一次進入一個事件子電路方塊時，需要執行某些操作。提供事件首次進入的標誌來辨別。

圖示



屬性

參數	描述
Event Subcircuit Block Name	標誌所給的事件子電路方塊名稱

標誌節點為一個輸出節點。當事件子電路方塊第一次進入時，節點值為 1，反之，為 0。舉例來說，當發現事件子電路方塊 S1 第一次進入時，設定 Event Subcircuit Block Name 為 S1。

全域變數

全域變數使用在條件敘述或特殊場合。

圖示

Global Variable



屬性

參數

描述

Name

全域變數名稱

Initial Value

全域變數初始值

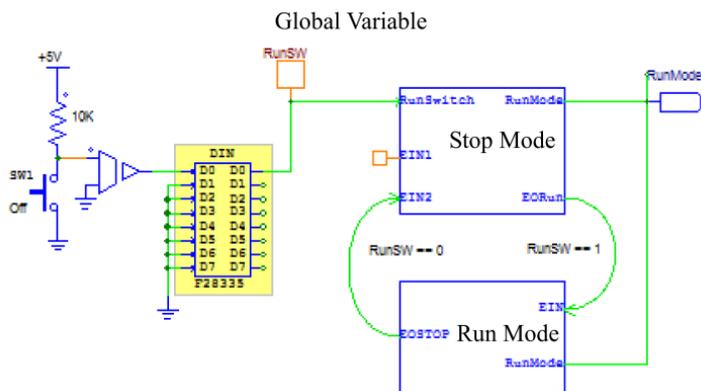
為了定義一個訊號為全域變數，連接全域變數元件到特定節點。注意只有用來產生程式碼的控制電路的訊號可定義為全域變數。

正如其名，全域變數可全範圍使用。當全域變數的初始值改變，在電路中，包含子電路的所有全域變數的初始值在同時都會改變。

全域變數可為訊號接收或訊號輸出，當為訊號接收端時，將從節點接收信號值，當為訊號輸出端時，會將值放在節點。

一旦在事件條件敘述中使用全域變數，所有在條件敘述中的變數須皆為全域變數。

舉例如下：



在此範例中，全域變數 *RunSW* 連接到數位輸入的輸出腳位 D0。全域變數使用在這兩個操作模式間轉換的條件敘述。

全域變數的另一用法為信號輸出，例如，全域變數可用來當作信號輸出與傳值至另一方塊。

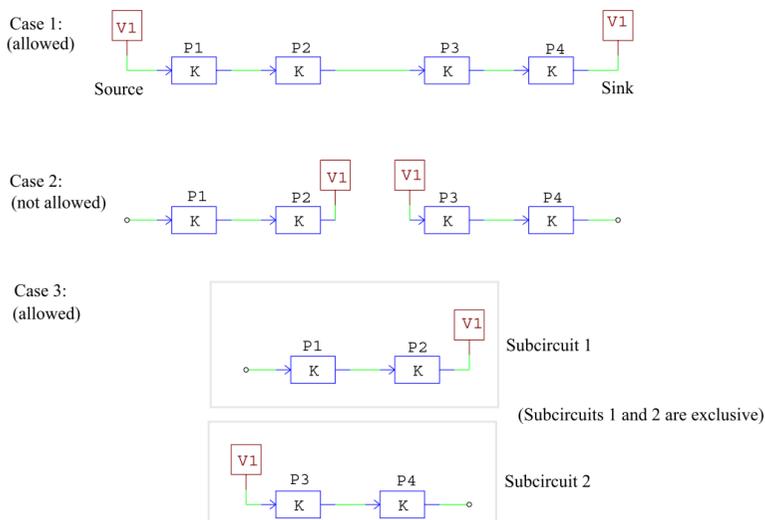


注意

全域變數不應用來當作從一節點傳值到另一節點的標籤。當兩個節點可經由線實際連接，全域變數的使用有下列限制：

- 只要同樣的傳輸路徑相同名稱的全域變數可多次使用。
- 如果不一樣的傳輸路徑，則不允許同樣名稱的全域變數，除非在不同的專屬狀態(不能同時發生的狀態為專屬狀態)。

為了說明這點，下面流程圖說明全域變數使用的場合。



在 Case 1，全域變數 V1 先為輸出源，並賦予方塊 P1 的輸入一個值。經過一連串計算，方塊 P4 的輸出也被賦予相同的全域變數 V1。因為兩個全域變數都在同一個傳輸路徑上，所以是可行的。

在 Case 2，全域變數 V1 用來當作從方塊 P2 的輸出傳值到方塊 P3 的輸入之標籤，這是不允許的。從一個節點傳值到另一個節點，應該用標籤取代或用線連接兩個節點。

在 Case 3，全域變數 V1 在子電路 S1 和 S2 都有使用，然而，子電路 S1 和 S2 為兩個專屬狀態，也就是說，系統將會執行子電路 1 或 2，並不會同時執行。在這個案例中全域變數的使用是允許的。

中斷

在硬體標的中，如數位輸入、編碼器、捕捉和 PWM 產生器(用於 F2833x 和 F2803x DSP)等元件可產生硬體中斷，中斷方塊允許使用者用對應的子電路聯繫產生中斷的元件來表示中斷服務程序。



注意

中斷元件不能置於子電路內，它只能位於在最上層的主電路。

圖示



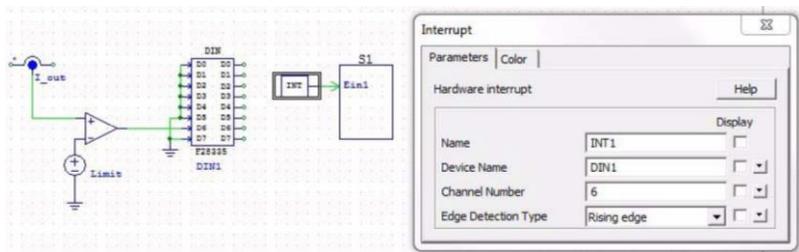
屬性

參數

描述

Device Name	<p>啟動硬體中斷的硬體設備名稱</p> <p>啟動中斷的設備之輸入通道數</p> <p>例如，數位輸入的通道 D0 產生中斷，通道數應設為 0。</p>
Channel Number	<p>注意此參數只使用在：</p> <ol style="list-style-type: none"> 1.數位輸入 2.捕捉(只在 PE-Expert3 標的) <p>並不適用於解碼器與 PWM 產生器。</p> <p>只用於數位輸入與捕捉，可為下列其中之一：</p> <p>No edge detection：不會發生中斷。</p> <p>Rising edge：輸入信號的上升緣產生中斷。</p> <p>Falling edge：輸入信號的下降緣產生中斷。</p> <p>Rising/Falling edges：輸入信號的上升與下降緣都會產生中斷。</p>
Trigger Type	<p>Falling edge：輸入信號的下降緣產生中斷。</p> <p>Rising/Falling edges：輸入信號的上升與下降緣都會產生中斷。</p>

下圖顯示如何使用中斷：



在這個電路中，量測電流 I_{out} 並與參考值 $Limit$ 比較，如果電流 I_{out} 超過 $Limit$ ，比較器的輸出將由 0 變為 1。數位輸入方塊 $DIN1$ 的通道 $D6$ 將接收到一上升邊緣，中斷方塊參數設定如圖所示：

1. Device Name： $DIN1$ 針對特定的數位輸入方塊
2. Channel Number：6 針對特定的數位輸入通道 $D6$
3. Edge Detection Type：針對 $I_{out} > Limit$ 條件下的上升邊緣

比較器輸出的上升邊緣將會產生硬體中斷且此操作將會經由輸入事件端口 $E11$ 轉換到子電路 $S1$ 。



注意

中斷方塊 $INT1$ 和事件子電路 $S1$ 間的連接為事件連接，並非一條線的連結。

附錄 E F2833x 硬體標的

有了 F2833x 硬體標的，SimCoder 可產生已準備好在基於 TI F2833x 浮點 DSP 的任何硬體板上執行的程式碼。

F2833x 硬體標的可在所有的 F2833x 封裝上執行，後兩頁的圖為 low-profile flatpack(LQFP)封裝的 F2833x DSP 之腳位定義。圖中實現 F2833x 硬體標的之主要函數以不同顏色來標注。

F2833x 硬體標的資料庫包含下列函數方塊：

- PWM 產生器(PWM generators)：三相(3-phase)、雙相(2-phase)、單相(1-phase)和 APWM
- PWM 產生器的開始/停止函數(Start/Stop functions for PWM generators)
- Trip-zone 與其狀態(Trip-zone and trip-zone state)
- 類比/數位轉換器(A/D Converter, ADC)
- 數位輸入與輸出(Digital input and output)
- SCI 組態、輸入與輸出(SCI configuration, input, output)
- SPI 組態、設備、輸入與輸出(SPI configuration, device, input, output)
- 捕捉與其狀態(Capture and capture state)
- 編碼器與其狀態(Encoder and encoder state)
- 上/下計數器(Up/Down counter)
- DSP 時脈(DSP clock)
- 硬體組態(Hardware configuration)

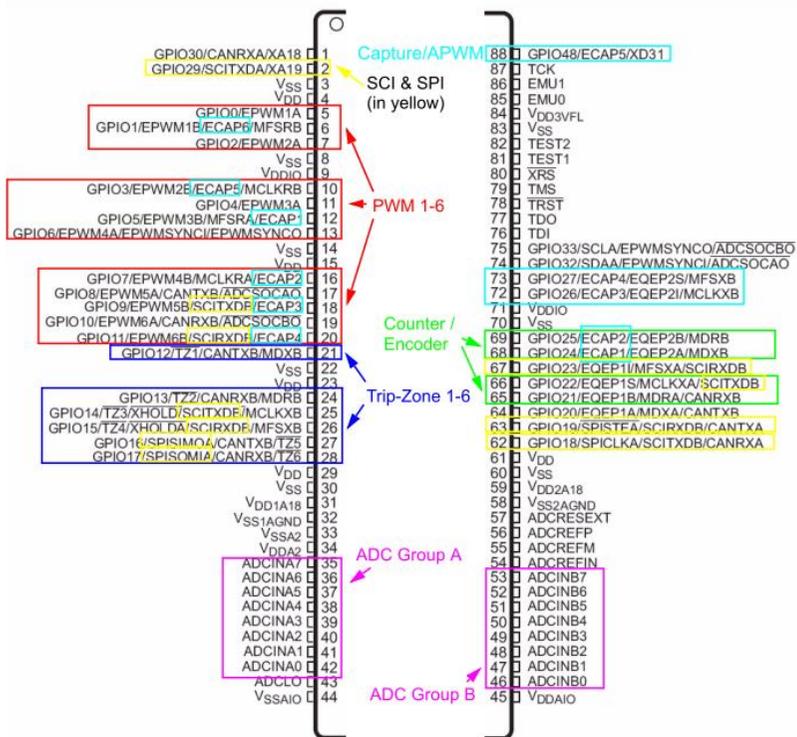
針對有多個取樣率的系統產生程式碼，SimCoder 使用 PWM 產生器的中斷為 PWM 取樣率。對於控制系統其他的取樣率，將先使用計數器 1 中斷，如需要再使用計數器 2 中斷，如果控制系統超過三個取樣率，相對應的中斷程序會在主程式中藉由軟體處理。

在 TI F2833x，PWM 產生器可以產生硬體中斷，SimCoder 將尋找與集結所有連接到 PWM 產生器並與其有相同取樣率的元件。這些元件將自動放置在所產生的程式碼的中斷服務程序中並執行。

此外，數位輸入、解碼器、捕捉和觸發區也可產生硬體中斷。每個硬體中斷必須有相對的中斷方塊(在 5.4 節中說明)，而每個中斷方塊則必須有相對應的中斷服務程序(表示中斷服務程序的子電路)。例如，PWM 產生器與數位輸入都產生中斷，它們各自都要有一個中斷方塊與中斷服務程序。

F2833x 硬體標的資料庫內元件定義將在此章說明。

F28335 DSP Port Assignments (Pin 1 - 88)



F28335 DSP Port Assignments (Pin 89 - 176)

132	GPIO75/XD4	GPIO76/XD3	133
131	GPIO74/XD5	GPIO77/XD2	134
130	GPIO73/XD6	GPIO78/XD1	135
129	GPIO72/XD7	GPIO79/XD0	136
128	GPIO71/XD8	GPIO38/XWE0	137
127	GPIO70/XD9	XCLKOUT	138
126	V _{DD}	V _{DD}	139
125	V _{SS}	V _{SS}	140
124	GPIO69/XD10	GPIO28/SCIRXDA/XZCS6	141
123	GPIO68/XD11	GPIO34/ECAP1/XREADY	142
122	GPIO67/XD12	V _{DDIO}	143
121	V _{DDIO}	V _{SS}	144
120	V _{SS}	GPIO36/SCIRXDA/XZCS0	145
119	GPIO66/XD13	V _{DD}	146
118	V _{SS}	V _{SS}	147
117	V _{DD}	GPIO35/SCITXDA/XRW	148
116	GPIO65/XD14	XRD	149
115	GPIO64/XD15	GPIO37/ECAP2/XZCS7	150
114	GPIO63/SCITXDC/XD16	GPIO40/XA0/XWE1	151
113	GPIO62/SCIRXDC/XD17	GPIO41/XA1	152
112	GPIO61/MFSRB/XD18	GPIO42/XA2	153
111	GPIO60/MCLKRB/XD19	V _{DD}	154
110	GPIO59/MFSRA/XD20	V _{SS}	155
109	V _{DD}	GPIO43/XA3	156
108	V _{SS}	GPIO44/XA4	157
107	V _{DDIO}	GPIO45/XA5	158
106	V _{SS}	V _{DDIO}	159
105	XCLKIN	V _{SS}	160
104	X1	GPIO46/XA6	161
103	V _{SS}	GPIO47/XA7	162
102	X2	GPIO80/XA8	163
101	V _{DD}	GPIO81/XA9	164
100	GPIO58/MCLKRA/XD21	GPIO82/XA10	165
99	GPIO57/SPISTEAXD22	V _{SS}	166
98	GPIO56/SPICLKA/XD23	V _{DD}	167
97	GPIO55/SPISOMIA/XD24	GPIO83/XA11	168
96	GPIO54/SPISIMOAXD25	GPIO84/XA12	169
95	GPIO53/EQEP1I/XD26	V _{DDIO}	170
94	GPIO52/EQEP1S/XD27	V _{SS}	171
93	V _{DDIO}	GPIO85/XA13	172
92	V _{SS}	GPIO86/XA14	173
91	GPIO51/EQEP1B/XD28	GPIO87/XA15	174
90	GPIO50/EQEP1A/XD29	GPIO39/XA16	175
89	GPIO49/ECAP6/XD30	GPIO31/CANTXA/XA17	176

SCI & SPI (in yellow)

Capture/APWM

Counter/Encoder

Capture/APWM

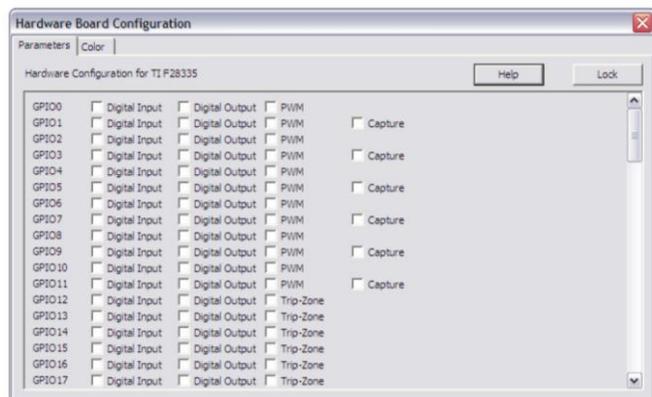
硬體配置

F2833x DSP 提供 88 個通用輸出入(general-purpose-input-output, GPIO)端口(GPIO0 到 GPIO87)，每個端口可能配置不同的功能。然而，對於一個特定的 DSP 板，並非每個端口皆可由外部設定且通常某些端口的功能是固定的。硬體配置方塊提供針對特定 DSP 板配置 SimCoder 的方式。

圖示

F28335
Board
Config

對話視窗如下圖所示：



對於每個 GPIO 提供選取可用函數的複選框，如選擇某一函數，就只能使用該函數，其餘的則無法使用。例如，GPIO1 可當數位輸入、數位輸出、PWM 和捕捉使用，一旦選用 GPIO1 來當 PWM 輸出，只能勾選 PWM 的複選框而其他的則須空白。如果在電路中 GPIO1 用來當數位輸入，則 SimCoder 會出現錯誤。

DSP 時脈

DSP 時脈方塊定義 F2833x DSP 的速度與外部時脈頻率以及程式空間大小。

圖示



屬性

參數	描述
External Clock(MHz)	DSP 板外部時脈的頻率，單位 MHz。此頻率必須為整數，且最大允許頻率為 30MHz。
DSP Speed(MHz)	DSP 速度，單位 MHz。需為整數且為外部時脈的整數倍，從 1 到 12 倍。最大允許速度為 150MHz。

如電路中不使用 DSP 時脈方塊，則使用 DSP 方塊的預設值。

PWM 產生器

F2833x DSP 提供六組 PWM 輸出：PWM 1(GPIO0 和 GPIO1), PWM 2(GPIO2 和 GPIO3), PWM 3(GPIO4 和 GPIO5), PWM 4(GPIO6 和 GPIO7), PWM 5(GPIO8 和 GPIO9)和 PWM 6(GPIO10 和 GPIO11)。每組有兩個彼此互補的輸出。舉例來說，除了當 PWM 操作在特殊模式外，PWM 1 有一個正輸出 PWM 1A 和一個負輸出 PWM 1B。

在 SimCoder 中，六個 PWM 可如下列方式使用：

- 兩個三相 PWM 產生器：PWM 123(包含 PWM 1, 2 和 3)和 PWM 456(包含 PWM 4, 5 和 6)。
- 六個雙相 PWM 產生器：在特殊操作模式下有兩個不互補輸出的 PWM 產生器 PWM 1, 2, 3, 4, 5 和 6。
- 單相 PWM 產生器：有著兩個互補輸出的 PWM 1, 2, 3, 4, 5 和 6。
- 相位偏移單相 PWM 產生器：有著兩個互補輸出的 PWM 2, 3, 4, 5 和 6。

這些 PWM 產生器可觸發 A/D 轉換器並使用 **trip-zone** 信號。

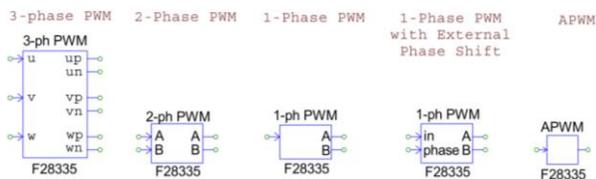
除了上述的 PWM 產生器，還有六個與捕捉使用同樣來源的 APWM 產生器。與六個 PWM 產生器(PWM1~6)相比，這些 PWM 產生器有不能觸發 A/D 轉換器與不能使用 **trip-zone** 信號的功能限制。因為共用來源，故端口做為捕捉使用時，就不能用來當 PWM 產生器。



SimCoder 中所有 PWM 產生器內部都包含一個切換週期的延遲，也就是說 PWM 產生器的輸入值在用來更新 PWM 輸出前會先延期一個週期。此延遲是模擬 DSP 硬體實現時的內部延遲，故其是必要的。

PWM 產生器的圖示與參數如下：

圖示



三相 PWM 產生器

三相 PWM 產生器的圖示中，“u”，“v”和“w”為三個相位(或稱為相位“a”，“b”和“c”)。字母“p”代表正輸出，“n”代表負輸出。例如，三相 PWM 123，“up”代表 PWM 1A，“un”代表 PWM 1B。

屬性

參數	描述
PWM Source	PWM 產生器的來源。 可為使用 PWM1~3 的"3-phase PWM 123" 或使用 PWM4~6 的"3-phase PWM 456"。
Dead Time	PWM 產生器的空白時間 Td，單位為秒。
Sampling Frequency	PWM 產生器的取樣頻率，單位 Hz。基於此頻率完成計算且更新 PWM 信號責任週期。
PWM Freq. Scaling Factor	PWM 頻率與取樣頻率間的比例因數，可為 1、2 或 3。亦即，PWM 頻率(用來控制開關切換的 PWM 輸出信號頻率)可為取樣頻率的倍數。例如，取樣頻率為 50kHz 且比例因數為 2，意謂著 PWM 頻率為 100kHz。開關以 100kHz 切換，但閘極信號每兩個切換週期 50kHz 更新一次。

Carrier Wave Type	<p>載波類型和初始 PWM 輸出狀態，為下列其一：</p> <p>Triangular(start low)：三角波，初始 PWM 輸出狀態為低準位。</p> <p>Triangular(start high)：三角波，初始 PWM 輸出狀態為高準位。</p> <p>Sawtooth(start low)：鋸齒波，初始 PWM 輸出狀態為低準位。</p> <p>Sawtooth(start high)：鋸齒波，初始 PWM 輸出狀態為高準位。</p>
Trigger ADC	<p>設定 PWM 產生器是否觸發 A/D 轉換器。為下列其一：</p> <p>Do not trigger ADC：PWM 不會觸發 A/D 轉換器。</p> <p>Trigger ADC Group A：PWM 將觸發 A/D 轉換器的 A 群組。</p> <p>Trigger ADC Group B：PWM 將觸發 A/D 轉換器的 B 群組。</p> <p>Trigger ADC Group A & B：PWM 將觸發 A/D 轉換器的 A 與 B 群組。</p>
ADC Trigger Position	<p>A/D 觸發位置的範圍從 0 到一個小於 1 的值。當為 0 時，A/D 轉換器從 PWM 週期開始時被觸發。當為 0.5 時，A/D 轉換器在 PWM 週期為 180°時被觸發。</p>
Use Trip-Zone i	<p>定義 PWM 產生器使用第 i 個 trip-zone 信號與否，i 的範圍從 1 到 6，為下列其一：</p> <p>Disable Trip-Zone i：停用第 i 個 trip-zone 信號。</p> <p>One shot：PWM 產生器在單次觸發模式下使用 trip-zone 信號。一旦觸發，PWM 須手動啟動。</p> <p>Cycle by cycle：PWM 產生器在週期循環的基礎下使用 trip-zone 信號。trip-zone 信號在當週內有效，PWM 將在下一個週期自動重新啟動。</p>

Trip Action	定義 PWM 產生器如何回應跳閘動作。為下列其一： High impedance：PWM 輸出為高阻抗。 PWM A high & B low：PWM 正輸出為高準位，負輸出為低準位。 PWM A low & B high：PWM 正輸出為低準位，負輸出為高準位。 No action：不採取任何動作。
Peak-to-peak Value	載波的峰對峰值 V_{pp} 。
Offset Value	載波的直流偏移量 V_{offset} 。
Initial Input Value u, v, w	三相輸入 u、v 和 w 的初始值。
Start PWM at Beginning	當設定為 "Start"，開始時 PWM 立即啟動。如果設定為 "Do not start"，則需使用 "Start PWM" 函數來啟動 PWM。

單相 PWM 產生器

帶有內部或外部相位偏移的單相 PWM 產生器具有相同的屬性，不同的是外部相位偏移的單相 PWM 有一個相位偏移的外部輸入(標記為 "phase")。相位的單位與沒有外部相位偏移的單相 PWM 相同都為角度。

屬性

參數	描述
PWM Source	PWM 產生器的來源。 沒有相位偏移，可為 PWM 1~PWM 6。 有相位偏移，可為 PWM 2~PWM 6。

Output Mode	<p>PWM 產生器的輸出模式，可為下列其一：</p> <p>Use PWM A&B：使用 PWM 輸出 A 和 B，為互補。</p> <p>Use PWM A：只使用 PWM 輸出 A。</p> <p>Use PWM B：只使用 PWM 輸出 B。</p>
Dead Time	<p>PWM 產生器的空白時間 T_d，單位為秒。</p>
Sampling Frequency	<p>PWM 產生器的取樣頻率，單位 Hz。基於此頻率完成計算且更新 PWM 信號責任週期。</p>
PWM Freq. Scaling Factor	<p>PWM 頻率與取樣頻率間的比例因數，可為 1、2 或 3。亦即，PWM 頻率(用來控制開關切換的 PWM 輸出信號頻率)可為取樣頻率的倍數。例如，取樣頻率為 50kHz 且比例因數為 2，意謂著 PWM 頻率為 100kHz。開關以 100kHz 切換，但閘極信號每兩個切換週期 50kHz 更新一次。</p>
Carrier Wave Type	<p>載波類型和初始 PWM 輸出狀態，為下列其一：</p> <p>Triangular(start low)：三角波，初始 PWM 輸出狀態為低準位。</p> <p>Triangular(start high)：三角波，初始 PWM 輸出狀態為高準位。</p> <p>Sawtooth(start low)：鋸齒波，初始 PWM 輸出狀態為低準位。</p> <p>Sawtooth(start high)：鋸齒波，初始 PWM 輸出狀態為高準位。</p>
Trigger ADC	<p>設定 PWM 產生器是否觸發 A/D 轉換器。為下列其一：</p> <p>Do not trigger ADC：PWM 不會觸發 A/D 轉換器。</p> <p>Trigger ADC Group A：PWM 將觸發 A/D 轉換器的 A 群組。</p> <p>Trigger ADC Group B：PWM 將觸發 A/D 轉換器的 B 群組。</p> <p>Trigger ADC Group A&B：PWM 將觸發 A/D 轉換器的 A 與 B 群組。</p>

ADC Trigger Position	A/D 觸發位置的範圍從 0 到一個小於 1 的值。當為 0 時，A/D 轉換器在 PWM 週期開始時被觸發。當為 0.5 時，A/D 轉換器在 PWM 週期為 180°時被觸發。
Use Trip-Zone i	<p>定義 PWM 產生器使用第 i 個 trip-zone 信號與否，i 的範圍從 1 到 6，為下列其一：</p> <p>Disable Trip-Zone i：停用第 i 個 trip-zone 信號。</p> <p>One shot：PWM 產生器在單次觸發模式下使用觸 trip-zone 信號。一旦觸發，PWM 須手動啟動。</p> <p>Cycle by cycle：PWM 產生器在週期循環的基礎下使用 trip-zone 信號。trip-zone 信號在當週內有效，PWM 將在下一個週期自動重新啟動。</p>
Trip Action	<p>定義 PWM 產生器如何回應跳閘動作。可為下列其一：</p> <p>High impedance：PWM 輸出為高阻抗。</p> <p>PWM A high & B low：PWM 正輸出為高準位，負輸出為低準位。</p> <p>PWM A low & B high：PWM 正輸出為低準位，負輸出為高準位。</p> <p>No action：不採取任何動作。</p>
Peak-to-peak Value	載波的峰對峰值 V_{pp} 。
Offset Value	載波的直流偏移量 V_{offset} 。
Phase Shift	相對於 PWM 產生器參考輸出的輸出相位偏移，單位度(°)。單相相移 PWM 針對此特性會多個輸入。
Initial Input Value	輸入的初始值。
Start PWM at Beginning	當設定為 "Start"，開始時 PWM 立即啟動。如果設定為 "Do not start"，則需使用 "Start PWM" 函數來啟動 PWM。

(8) 相位偏移

單相 PWM 產生器可以產生一個相對另一 PWM 信號有相位偏移的 PWM 信號。兩個 PWM 系列：PWM 1, 2, 3 與 PWM 4, 5, 6, 敘述如下：

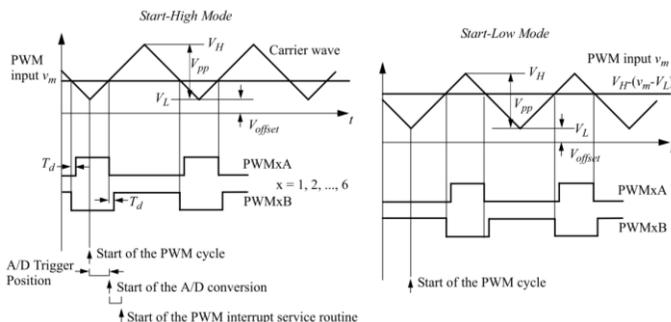
- 參考 PWM 與相位偏移 PWM 需來自同一系列，也就是 PWM 1 為參考，PWM 2 和 3 或 PWM 4, 5 和 6 可相對 PWM 1 做相位偏移。或 PWM 2 為參考，PWM 3 可相對 PWM 2 做相位偏移。相同的，PWM 4(或 5)為參考，PWM 5(或 6)相對 PWM 4(或 5)做相位偏移。但使用 PWM 2(或 3)當 PWM 4, 5 和 6 相位偏移的參考則不允許。
- 參考 PWM 和相位偏移 PWM 在系列中需是連續的，不能使用 PWM1 當 PWM3 的相位偏移參考且不能讓 PWM2, 5 或 6 知道。

相位偏移值的單位為角度，當值為 -30° ，就會相對參考 PWM 產生器的輸出在一個切換週期中向右(落後)偏移 30° ，其等效於 PWM 載波向右偏移 30° 。當相位值為 30° ，則輸出為向左(領先)偏移 30° 。

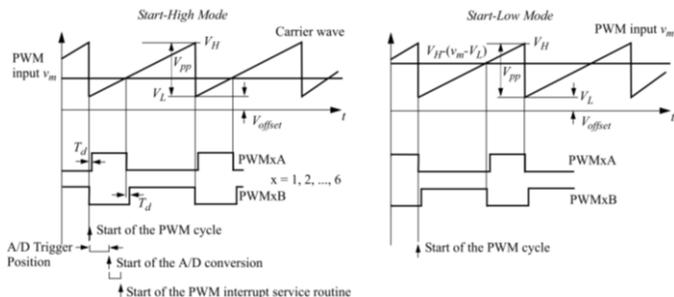
(9) 載波

載波波形有兩種類型：三角波(有等效上升和下降斜率區間)和鋸齒波。此外，也有兩種操作模式：低啟動和高啟動模式，說明如下：

三角波載波的 PWM 產生器輸入和輸出波形如下：



鋸齒波載波的 PWM 產生器輸入和輸出波形如下：



上圖說明如何定義空白時間和當 PWM 產生器觸發 A/D 轉換器的時間序列。如果選擇觸發 A/D 轉換器，從 PWM 週期開始，在某個由 A/D 轉換器觸發位置定義的延遲時間後，A/D 轉換將開始動作。在 A/D 轉換完成後，PWM 中斷服務程序將會開始。

如果 PWM 產生器不觸發 A/D 轉換器，PWM 中斷服務程序將在 PWM 週期開始時執行。

上圖顯示高啟動與低起動模式如何工作。假定 PWM 輸入 v_m 與載波最低值 V_L 和最高值 V_H 。在高啟動模式，PWM 正輸出 PWMA 在切換週期開始時為高準位，只要輸入 v_m 大於載波則一直維持在高準位。例如，當載波由 0 到 1， $V_L=0$ 且 $V_H=1$ ，假設 $v_m=0.2$ ，只要載波小於 0.2 則 PWM 輸出 PWMA 將維持在高準位。

另一方面，在低啟動模式，PWM 正輸出 PWMA 在切換週期開始時為低準位，當載波大於 $V_H - (v_m - V_L)$ 時則為高準位。例如，當載波由 0 到 1， $V_L=0$ 且 $V_H=1$ ，

假設 $v_m=0.2$ ，只要載波大於 0.8 則 PWM 輸出 PWMA 將維持在高準位。



注意

低起動模式中，在與載波比較產生 PWM 信號之前，PWM 輸入 v_m 在內部被轉換到 $V_H - (v_m - V_L)$ 。在這轉換中，高啟動和低起動模式有著相同的責任週期表示。例如，對於一個 $V_L=0$ 以及 $V_H=1$ 的鋸齒波或是一個 $V_L=-V_H$ 的三角波，高啟動和低起動的 PWMA 輸出的責任週期 D 皆為 $D = v_m / V_H$ 。

雙相 PWM 產生器

屬性

參數	描述
PWM Source	PWM 產生器的來源。可為 PWM 1~PWM 6。
Mode Type	PWM 產生器的操作模式。為 6 個模式之一，6 個操作模式的波形說明如下：
Sampling Frequency	PWM 產生器的取樣頻率，單位 Hz。基於此頻率完成計算且更新 PWM 信號責任週期。
PWM Freq. Scaling Factor	PWM 頻率與取樣頻率間的比例因數，可為 1、2 或 3。亦即，PWM 頻率(用來控制開關切換的 PWM 輸出信號頻率)可為取樣頻率的倍數。例如，取樣頻率為 50kHz 且比例因數為 2，意謂著 PWM 頻率為 100kHz。開關以 100kHz 切換，但閘極信號每兩個切換週期 50kHz 更新一次。
Trigger ADC	<p>設定 PWM 產生器是否觸發 A/D 轉換器。為下列其一：</p> <p>Do not trigger ADC：PWM 不會觸發 A/D 轉換器。</p> <p>Trigger ADC Group A：PWM 將觸發 A/D 轉換器的 A 群組。</p> <p>Trigger ADC Group B：PWM 將觸發 A/D 轉換器的 B 群組。</p> <p>Trigger ADC Group A&B：PWM 將觸發 A/D 轉換器的 A 與 B 群組。</p>
ADC Trigger Position	A/D 觸發位置的範圍從 0 到一個小於 1 的值。當為 0 時，A/D 轉換器在 PWM 週期開始時被觸發。當為 0.5 時，A/D 轉換器在 PWM 週期為 180°時被觸發。

Use Trip-Zone i	<p>定義 PWM 產生器使用第 i 個 trip-zone 信號與否，i 的範圍從 1 到 6，為下列其一：</p> <p>Disable Trip-Zone i：停用第 i 個 trip-zone 信號。</p> <p>One shot：PWM 產生器在單次觸發模式下使用 trip-zone 信號。一旦觸發，PWM 須手動啟動。</p> <p>Cycle by cycle：PWM 產生器在週期循環的基礎下使用 trip-zone 信號。trip-zone 信號在當週內有效，PWM 將在下一個週期自動重新啟動。</p>
Trip Action	<p>定義 PWM 產生器如何回應跳閘動作。可為下列其一：</p> <p>High impedance：PWM 輸出為高阻抗。</p> <p>PWM A high & B low：PWM 正輸出為高準位，負輸出為低準位。</p> <p>PWM A low & B high：PWM 正輸出為低準位，負輸出為高準位。</p> <p>No action：不採取任何動作。</p>
Peak Value	載波的峰值 V_{pk} 。
Initial Input Value A, B	輸入 A 和 B 的初始值。
Start PWM at Beginning	當設定為 "Start"，開始時 PWM 立即啟動。如果設定為 "Do not start"，則需使用 "Start PWM" 函數來啟動 PWM。

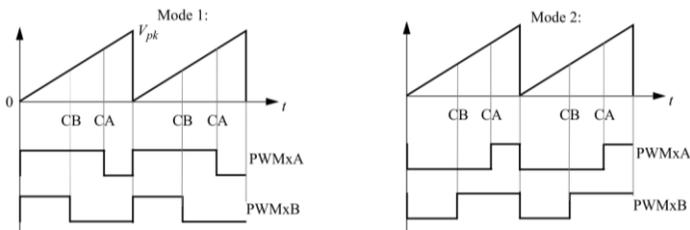
對於雙相 PWM 產生器，輸出是依據操作模式來決定，如下所示。載波依據操作模式可為鋸齒波或三角波，從 0 增加到峰值 V_{pk} ，且無偏移。

(10) 操作模式 1

下圖左為模式 1 的波形。如圖 "CA" 和 "CB" 指雙相 PWM 產生器的兩個輸出 A 和 B。每個輸入控制每個輸出的截止時間。

(11) 操作模式 2

下圖右為模式 2 的波形。不像模式 1，每個輸入控制每個輸出的導通時間。

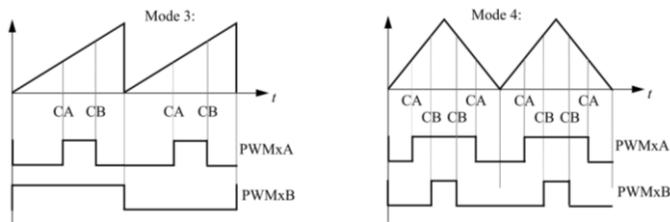


(12) 操作模式 3

下圖左為模式 3 的波形。輸入 A 控制 PWM 輸出 A 的導通時間且輸入 B 控制截止時間。PWM 輸出 B 導通時為一完整的 PWM 週期，在下一個週期截止。

(13) 操作模式 4

下圖右為模式 4 的波形。載波為三角波，每個輸入控制其輸出的導通與截止。

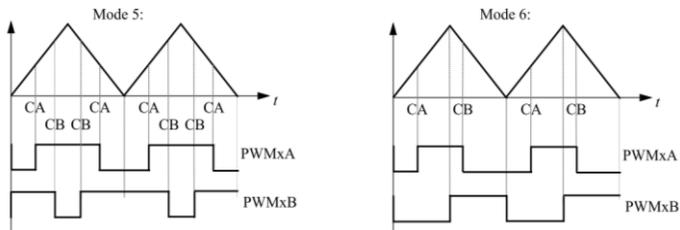


(14) 操作模式 5

下圖左為模式 5 的波形。載波為三角波，與模式 4 相似，每個輸入控制其輸出的導通與截止，注意在這例子中 PWM 輸出 B 為交錯。

(15) 操作模式 6

下圖右為模式 6 的波形。輸入 A 控制 PWM 輸出 A 的導通時間且輸入 B 控制截止時間，PWM 輸出 B 在前半週為導通，在後半週為截止。



APWM 產生器

屬性

參數	描述
PWM Source	<p>APWM 產生器與捕捉使用相同來源。PWM 來源可為使用 14 個指定 GPIO 端口的 6 個 APWM 的其中一個，如下所列：</p> <p>APWM1 (GPIO5, GPIO24, GPIO34) APWM2 (GPIO7, GPIO25, GPIO37) APWM3 (GPIO9, GPIO26) APWM4 (GPIO11, GPIO27) APWM5 (GPIO3, GPIO48) APWM6 (GPIO1, GPIO49)</p>
PWM Frequency	PWM 產生器的頻率，單位 Hz。
Carrier Wave Type	<p>載波類型和初始 PWM 輸出狀態，為下列其一：</p> <p>Sawtooth(start low)：鋸齒波，初始 PWM 輸出狀態為低準位。</p> <p>Sawtooth(start high)：鋸齒波，初始 PWM 輸出狀態為高準位。</p>
Stop Action	<p>當 PWM 產生器停止時的輸出狀態。可為下列其中之一：</p> <p>Output low：PWM 輸出設定為低準位。</p> <p>Output high：PWM 輸出設定為高準位。</p>

Peak-to-peak Value	載波的峰對峰值。
Offset Value	載波的直流偏移量。
Phase Shift	相對於 PWM 產生器參考輸出的輸出相位偏移，單位度(°)。
Initial Input Value	輸入初始值。
Start PWM at Beginning	當設定為"Start"，開始時 PWM 立即啟動。如果設定為"Do not start"，則需使用"Start PWM"函數來啟動 PWM。

相似於單相 PWM 產生器，APWM 產生器可產生一個參照另一個 PWM 產生器的相位偏移 PWM 信號。相位偏移規則如同單相 PWM 產生器。

如先前提醒，APWM 產生器相較於單相 PWM 產生器減少一些功能，不能觸發 A/D 轉換器也無法使用 trip-zone 信號。

PWM 開始與停止

PWM 開始與 PWM 停止方塊提供開始或停止 PWM 產生器的功能，圖示與屬性如下所示：

圖示



屬性

參數	描述
PWM Source	PWM 產生器的來源。可為 PWM 1-6、三相 PWM 123 和 PWM 456 以及捕捉 1-6。

觸發區與其狀態

F2833x DSP 內提供了 6 個 trip-zone，trip-zone 1 到 6 使用 GPIO12 到 GPIO17 端口，trip-zone 是用來處理外部故障或是跳閘條件，相對應的 PWM 輸出可以透過編程做出對應的動作。

一個 trip-zone 信號可以被多個 PWM 產生器使用，一個 PWM 產生器可使用任何一個或是 6 個信號信號，信號信號產生的中斷可由中斷方塊處理。

當輸入信號為低準位時，trip-zone 信號會觸發跳閘動作。

圖示



屬性(trip-zone)

參數	描述
Port GPIO12 as Trip-Zone 1	定義若使用 GPIO12 端口為 trip-zone 1。
Port GPIO13 as Trip-Zone 2	定義若使用 GPIO13 端口為 trip-zone 2。
Port GPIO14 as Trip-Zone 3	定義若使用 GPIO14 端口為 trip-zone 3。
Port GPIO15 as Trip-Zone 4	定義若使用 GPIO15 端口為 trip-zone 4。
Port GPIO16 as Trip-Zone 5	定義若使用 GPIO16 端口為 trip-zone 5。
Port GPIO17 as Trip-Zone 6	定義若使用 GPIO17 端口為 trip-zone 6。

屬性(trip-zone 狀態)

參數	描述
PWM Source	PWM 產生器的來源。可為 PWM1-6、三相 PWM 123 和 PWM 456。

可在單次觸發模式或週期循環模式下產生 trip-zone 中斷，在 PWM 產生器參數輸入下定義。週期循環模式下中斷只會影響當前的 PWM 輸出，另一方面，單次觸發模式下，當輸入信號為低準位時中斷觸發跳閘動作。會將 PWM 永久設置，PWM 必須重新啟動才能恢復操作。

當觸發 PWM 產生器產生中斷時，trip-zone 狀態元件表示 trip-zone 信號處於單次觸發模式或週期循環模式下，當輸出為 1，意謂著 trip-zone 信號處於單次觸發模式下，當輸出為 0，trip-zone 信號處於週期循環模式下。

在這例子中 PWM 輸出 B 為交錯。



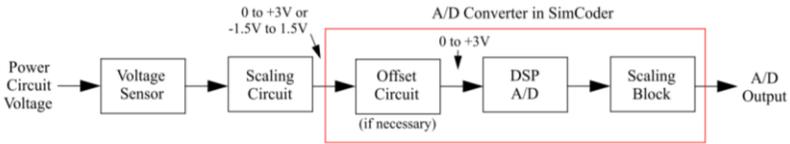
當定義對應 trip-zone 的中斷方塊，中斷方塊的參數“Device Name”應為 PWM 產生器的名稱，而非 trip-zone 方塊的名稱。例如，若 PWM 產生器“PWM_G1”使用 trip-zone 方塊“TZ1”的 trip-zone1。對應中斷方塊的名稱“Device Name”應為“PWM_G1”而非“TZ1”。中斷方塊的參數“Channel Number”在這個例子中沒有使用。

數位/類比轉換器

F2833x DSP 的 12 位元、16 個通道的數位/類比轉換器(ADC)可分成兩個區塊：Group A 及 Group B。在 DSP 上 A/D 轉換器之電壓輸入範圍為 0~+3V。

通常功率級電路的物理量(電壓、電流、速度等)會先經過幾級電路後再送入 DSP，例如：功率級電路上的電壓準位通常比較高，首先使用電壓感測器轉換成控制信號，再由比例運算電路將信號縮放，如有需要再由一個偏移量電路提供信號直流偏移量，所以進到 DSP A/D 輸入的信號都在 0~+3V 間。此信號在 DSP 內轉換成數位值且運用比例運算方塊還原回其原始值。

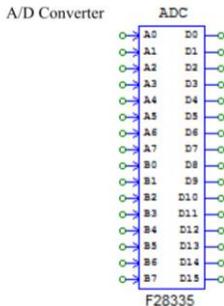
完整過程如下圖所示：



如上圖，SimCoder 中的 A/D 轉換器元件不完全與實際 DSP 的相同。更明確的說，它包含偏移量電路、A/D 轉換器與比例運算方塊。這樣的設計便於交流系統的應用。

SimCoder 資料庫中 A/D 轉換器的圖示與屬性說明如下。往後說明“ A/D 轉換器 ”指的是 SimCoder 資料庫中的 A/D 轉換器，而非 DSP 的，除非另有說明。

圖示



屬性

參數	描述
ADC Mode	<p>定義 A/D 轉換器的操作模式。為下列其一：</p> <p>Continuous：A/D 轉換器連續地執行轉換。當讀取轉換器的值時，其為最後一次轉換的結果。</p> <p>Start/stop(8-channel)：A/D 轉換器只依據請求執行轉換，只在一個 8 通道群組。</p> <p>Start/stop(16-channel)：A/D 轉換器只依據請求執行轉換，在所有的 16 個通道。</p>
Ch Ai or Bi Mode	<p>A/D 轉換器通道 Ai 或 Bi 的輸入模式，i 從 0 到 7。輸入模式可為下列其一：</p> <p>AC：輸入範圍從 -1.5V 到 +1.5V。此選項包含進入 A/D 轉換器的偏移電路，當需要外部準位調節器調整交流信號至 0~+3V 的範圍之情形時是方便的。</p> <p>DC：輸入為一直流值，範圍為 0~+3V。</p>
Ch Ai or Bi Gain	A/D 轉換器通道 Ai(或 Bi)的增益 k, i 從 0 到 7。

操作模式

當設定為連續模式時，A/D 轉換器可自主地執行轉換。PWM 產生器觸發“啟動/停止”模式的轉換。



注意

PWM 產生器觸發 A/D 轉換器有下列的限制：

- A/D 轉換器只能被一個 PWM 產生器觸發。也就是如果有多個 PWM 產生器，只有一個可設定用來觸發 A/D 轉換器，其餘的應設定不能觸發 A/D 轉換器。
- 當群組內某些信號也在電路內但與 PWM 產生器的頻率有不同的取樣率，則不允許 A/D 轉換器由一個 PWM 產生器觸發。在這樣情況下，建議 A/D 轉換器設為連續模式。

輸出縮放

輸出依據下列方程式作縮放：

$$V_o = k * V_i$$

其中 V_i 為 A/D 轉換器輸入端口的值。

輸入偏移與縮放

注意 A/D 轉換器的輸入必須保持在輸入範圍內，當輸入超出範圍時，會被箝位在極限值並給一個警告信息。

A/D 轉換器輸入端口的信號也必須如此縮放，當輸入通道模式為直流時，最大輸入電壓將被縮放到+3V；當輸入通道模式為交流時，最大輸入峰值電壓將被縮放到+1.5V。

在許多應用中，其電路監控的變數為交流信號，特別是交流馬達驅動系統。對於每個這類的交流信號，為了要調整信號準位到可被接受的 0 到+3V 範圍，硬體電路須在 DSP 類比輸入端加入偏移電路。

對於這種情況，F2833x DSP 的 SimCoder A/D 轉換器提供一個便利的方式。取代準位調整與 A/D 輸出信號縮放，使用者可在 SimCoder A/D 轉換器中選擇偏移選項與比例因數，將會產生相對應的標的程式碼。

下列有兩個範例說明如何使用 A/D 轉換器：一個直流輸入而另一個交流輸入。

(16) A/D 轉換器通道直流模式範例

假設功率級迴路電壓為一直流量，範圍如下：

$$V_{i_min} = 0V, V_{i_max} = 150V$$

A/D 轉換器的輸入模式設定為直流，範圍從 0 到 3V。假設在某一個點電壓的實際值為：

$$V_i = 100V$$

設定電壓取樣增益為 0.01，經過電壓取樣後，輸入的最大值與實際值變為：

$$V_{i_max_s} = 150 * 0.01 = 1.5V$$

$$V_{i_s} = 100 * 0.01 = 1V$$

為了運用 DSP 的全部範圍，使用一個增益為 2 的調節電路。結合電壓取樣的增益與調節電路後變成： $0.01 * 2 = 0.02$ 。經過調節電路後，DSP A/D 轉換器的輸入最大值和實際值變為：

$$V_{i_max_s_c} = 1.5 * 2 = 3V$$

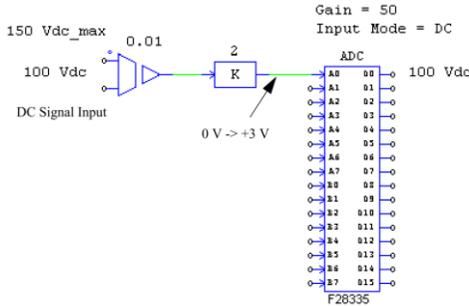
$$V_{i_s_c} = 1 * 2 = 2V$$

選擇在 DSP A/D 轉換器後的縮放方塊可恢復到原本功率級電路的數值，在此範例中，增益為 50，為調節電路與電壓感測器增益組合的倒數，A/D 轉換器的輸出最大值與實際值為：

$$V_{o_max} = 50 * 3 = 150V$$

$$V_o = 50 * 2 = 100V$$

PSIM 中 A/D 通道的增益設定為 50，電路連接與設定如下圖所示：



注意

此範例中如果比例方塊的增益由 2 變為 1，則 A/D 增益由 50 變為 100，模擬結果將會相同，但產生的硬體程式碼則不正確。這是因為硬體程式碼以為的最大輸入值會被縮放到+3V，但此範例只到 1.5V。因此，在直流模式中，使用者必須這樣設定電路使得最大的輸入值縮放到+3V。

A/D 轉換器通道交流模式範例

在另一個範例，假設功率級迴路電壓為一交流量，範圍如下：

$$V_{i_max} = +/- 75V$$

A/D 轉換器的輸入模式設定為交流，範圍從-1.5V 到+1.5V。假設電壓的實際值其峰值為：

$$V_i = +/- 50V$$

設定電壓取樣增益為 0.01，經過電壓取樣後，輸入的最大值與實際值變為：

$$V_{i_max_s} = +/- 0.75V$$

$$V_{i_s} = +/- 0.5V$$

因為 A/D 轉換器的輸入範圍從-1.5V 到+1.5V，在信號送進 DSP 前需要做縮放，需要一個增益為 2 的調節電路(亦即 $1.5/0.75=2$)。經過調節電路後，DSP A/D 轉換器的輸入最大值和實際值變為：

$$V_{i_max_s_c} = +/- 1.5V$$

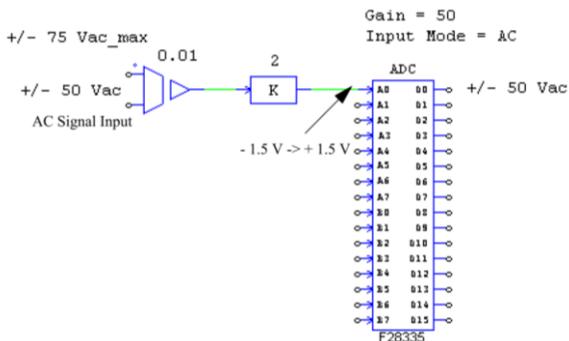
$$V_{i_s_c} = +/- 1V$$

選擇在 DSP A/D 轉換器後的縮放方塊可恢復到原本功率級電路的數值，在此範例中，增益為 50，為調節電路與電壓感測器增益組合的倒數，A/D 轉換器的輸出最大值與實際值為：

$$V_{o_max} = +/- 75V$$

$$V_o = +/- 50V$$

PSIM 中 A/D 通道的增益設定為 50，電路連接與設定如下圖所示



注意

在此電路中，交流信號直接傳送給 A/D 轉換器。這是因為當 A/D 輸入設定為交流時，輸入範圍從-1.5V 到+1.5V，A/D 轉換器方塊已經包含執行直流偏移的調節電路之功能。在實際硬體電路，交流信號仍須經由縮放與偏移以達到 DSP A/D 轉換器所需的範圍 0V 到 3V 內。

確保針對硬體所產生的程式碼之正確性，在 A/D 轉換器輸入端口的最大峰值必須縮放到 1.5V。

數位輸出與輸入

F2833x DSP 有 88 個可配置為數位輸入(Digital Input)或數位輸出(Digital Output)的通用輸出入(general-purpose-input-output, GPIO)端口。SimCoder 提供了八通道的數位輸出入方塊，多個八通道方塊可在同一個電路中使用。

圖示



屬性(數位輸入)

參數	描述
Port Position for Input i	輸入 i 的端口位置，i 從 0 到 7。為 88 個 GPIO 端口的其中之一，從 GPIO0 到 GPIO87。
Use as External Interrupt	顯示若此端口用來當外部中斷輸入。

屬性(數位輸出)

參數	描述
Port Position for Output i	輸出 i 的端口位置，i 從 0 到 7。為 88 個 GPIO 端口的其中之一，從 GPIO0 到 GPIO87。
 注意	當 GPIO 作為輸入端口時，同一端口就不可當外設端口使用。例如，GPIO1 設定為數位輸入也同時為 PWM1 輸出時，就會回報錯誤。

在 F2833x DSP 中，從 GPIO0 到 GPIO63 最多可定義七個外部中斷源 (具體而言，從 GPIO0 到 GPIO31 最多 2 個，GPIO32 到 GPIO63 最多 5 個)。GPIO0 到 GPIO31 的外部中斷比 GPIO32 到 GPIO63 的優先。

上/下計數器

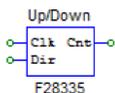
F2833x DSP 有兩個上/下計數器。第一個可為 GPIO20-21 或 GPIO50-51，第二個則在 GPIO24-25。



注意

在 GPIO20-21 與 GPIO50-51 的計數器 1 使用相同的內部函數方塊，但不可同時使用。

圖示



屬性

參數

描述

參數	描述
Counter Source	計數器的來源。為下列其一： Counter1(GPIO20, 21)：計數器 1 使用 GPIO20 和 21。 Counter1(GPIO50, 51)：計數器 1 使用 GPIO50 和 51。 Counter2(GPIO24, 25)：計數器 1 使用 GPIO24 和 25。

在圖示中，“Clk”為輸入時脈信號，“Dir”為計數方向的信號。當 Dir 輸入為 1，計數器將正向計數，當輸入為 0 時，計數器將反向計數。



注意

“Clk”輸入對應計數器的第一個端口，“Dir”輸入對應第二個。例如，當計數器 1 為 GPIO20-21 時，GPIO20 為“Clk”而 GPIO21 為“Dir”。

上/下計數器的輸出給予計數值。



注意

上/下計數器與編碼器使用相同的來源。但同一個 GPIO 不能同時為計數器和編碼器。例如，同時使用編碼器 1 和計數器 1 會產生衝突解不被允許。

編碼器與其狀態

F2833x DSP 有兩個編碼器，編碼器 1 可為 GPIO20-21 或 GPIO50-51，編碼器 2 為 GPIOP24-25。



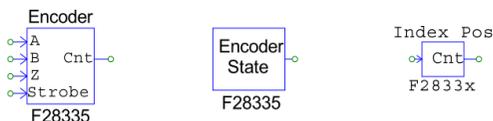
注意

在 GPIO20-21 與 GPIO50-51 的編碼器 1 使用相同的內部函數方塊，但不可同時使用。編碼器的輸出給予計數值。

編碼器狀態方塊是用來表示哪個輸入信號(或指示信號或選通信號)產生中斷。也可藉由指示(Z)信號與選通信號產生硬體中斷而編碼器狀態的輸出將表示何種信號產生中斷。當輸出為 0 即指示信號產生中斷，當輸出為 1 即選通信號產生中斷。

編碼器指示/選通位置方塊用來鎖住編碼器的初始位置。當此方塊輸入為 0 時，編碼器計數將設定為 0，當輸入變為 1 時編碼器開始動作，當指示/選通事件發生時編碼器會鎖住計數值。

圖示



屬性(編碼器)

參數	描述
Encoder Source	<p>編碼器的來源。為下列其一：</p> <p>Encoder1 (GPIO20, 21)：編碼器 1 使用 GPIO20 和 21，GPIO22 為選通而 GPIO23 為指示(Z)。</p> <p>Encoder1 (GPIO50, 51)：編碼器 1 使用 GPIO50 和 51，GPIO52 為選通而 GPIO53 為指示(Z)。</p> <p>Encoder2 (GPIO24, 25)：編碼器 2 使用 GPIO24 和 25，GPIO27 為選通而 GPIO28 為指示(Z)。</p>
Use Z Signal	定義若編碼器使用指示(Z)信號。

Use Strobe Signal	定義若編碼器使用選通信號。
Counting Direction	計數方向可為正向或反向。當設定為正向，計數器遞增計數，反之，計數器遞減計數。
Z Signal Polarity	定義指示信號的觸發極性： 高準位動作 低準位動作
Strobe Signal Polarity	定義選通信號的觸發極性： 高準位動作 低準位動作
Encoder Resolution	外部編碼器硬體的解析度。當為 0 時，編碼計數器會持續計數且不會重置，例如，解析度設置為 4096，當到達 4095 時，計數器將重置為 0。

屬性(編碼器狀態)

參數	描述
Encoder Source	定義哪個編碼器產生中斷。為下列其一，需為同一原理圖上的同一編碼器： Encoder1(GPIO20, 21)：編碼器 1 使用 GPIO20 和 21。 Encoder1(GPIO50, 51)：編碼器 1 使用 GPIO50 和 51。 Encoder2(GPIO24, 25)：編碼器 2 使用 GPIO24 和 25。

屬性(編碼器指示/選通位置)

參數	描述
Encoder Source	<p>定義哪個編碼器產生中斷。為下列其一，需為同一原理圖上的同一編碼器：</p> <p>Encoder1 (GPIO20, 21)：編碼器 1 使用 GPIO20 和 21。</p> <p>Encoder1 (GPIO50, 51)：編碼器 1 使用 GPIO50 和 51。</p> <p>Encoder2 (GPIO24, 25)：編碼器 2 使用 GPIO24 和 25。</p>
Latch Position	<p>指定保持計數形式，由下列選擇：</p> <p>IndexPos，若編碼器設定"Use Z Signal"不為"No"。</p> <p>StrobePos，若編碼器設定"Use Strobe Signal"不為"No"。</p>
Type of Postion	<p>可由下列做選擇：</p> <p>第一次鎖住的位置，或</p> <p>目前鎖住的位置</p>

捕捉與其狀態

F2833x DSP 提供六個捕捉。捕捉可產生中斷，且中斷觸發模式將由中斷方塊定義。

圖示



捕捉狀態方塊輸出為 1 或 0, 1 表示為上升邊緣而 0 表示為下降邊緣。
屬性(捕捉)：

參數	描述
Capture Source	捕捉的來源。使用 14 個指定 GPIO 的六個捕捉，如下所列： Capture1 (GPIO5, GPIO24, GPIO34) Capture2 (GPIO7, GPIO25, GPIO37) Capture3 (GPIO9, GPIO26) Capture4 (GPIO11, GPIO27) Capture5 (GPIO3, GPIO48) Capture6 (GPIO1, GPIO49)
Event Filter	事件濾波器預分頻。輸入信號由選定的預分頻劃分。
Timer Mode	捕捉計數器定時模式。可為 Absolute time 或 Time Difference。

屬性(捕捉狀態)：

參數	描述
Capture Source	捕捉的來源。可為 6 個捕捉的其中之一。

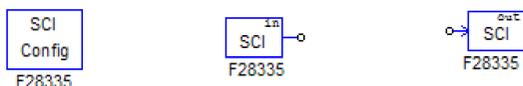
串列傳輸界面(SCI)

F2833x DSP 提供串列傳輸界面(SCI)的功能。透過 SCI，DSP 內部數據可由外部 RS-232 線傳送到電腦端。PSIM 提供 DSP 與電腦兩端所有傳送與接收數據必要的功能並在電腦端顯示。這是一個便於即時監控、除錯與調整 DSP 程式碼的方式。

對於 SCI 與監控功能更詳細的描述請參考“Tutorial - Using SCI for Real-time Monitoring in F2833x Target.pdf”文件。

SimCoder 提供三個 SCI 函數方塊：SCI 配置、SCI 輸入與 SCI 輸出說明如下：

圖示



SCI 配置

SCI 配置方塊定義 SCI 端口、傳輸速度、同位元檢查類型與數據緩衝區大小。

屬性

參數	描述
SCI Port	定義 SCI 端口，有 7 組 GPIO 端口可供 SCI 使用，如下所列： SCIA(GPIO28, GPIO29) SCIA(GPIO35, GPIO36) SCIB(GPIO9, GPIO11) SCIB(GPIO14, GPIO15) SCIB(GPIO18, GPIO19) SCIB(GPIO22, GPIO23) SCIC(GPIO62, GPIO63)

Speed (bps)	SCI 傳輸速度，單位 bps(每秒多少位元)，目前所提供的速度列表：200000, 115200, 57600, 38400, 19200 和 9600bps。或手動指定任何其他速度。
Parity Check	在傳輸中針對誤檢查的同位元檢查設定。可為無、奇或偶。
Output Buffer Size	針對 SCI 在 DSP 中配置的數據緩衝區大小，緩衝區位於 RAM 內，每個緩衝區元件儲存一包含三個 16 位元的字(6 位元組或 48 位元)的資料點。
 注意	緩衝區大小需適當選擇。一方面，為了收集更多數據以便長時間監控更多變數故偏好大的緩衝區，另一方面，DSP 內部記憶體有限，緩衝區太大則會干擾 DSP 的正常運作。

對於如何選擇緩衝區大小，更詳盡的資訊請參照“Tutorial - Using SCI for Real-time Monitoring in F2833x Target.pdf”文件。

SCI 輸入

SCI 輸入方塊用來定義一可變 DSP 程式碼中的變數。SCI 輸入變數的名稱會顯示在 DSP 示波器(在 Utilities 選單下)中，在執行時可透過 SCI 改變其數值。

SCI 輸入方塊提供一個便捷的方式來改變參考值或微調控制器參數。

屬性

參數	描述
Initial value	SCI 輸入變數的初始值

在原理圖中，SCI 輸入可視為常數。程式碼在 DSP 中執行時儘管其值可被改變，但模擬時還是被固定在初始值。

SCI 輸出

SCI 輸出方塊用來定義變數的顯示。當 SCI 輸出方塊連接到一個節點，SCI 輸出方塊的名稱會顯示在 DSP 示波器(在 Utilities 選單下)中，在執行期間，此變數的數據將透過 SCI 由 DSP 傳送到電腦，在 DSP 示波器上顯示波形。

SCI 輸出方塊提供一個方便監控 DSP 波形的的方法。

屬性

參數	描述
Data Point Step	定義如何收集常見數據。如果 Data Point Step 為 1，每個數據將被收集與傳送。如果設定為 10，每 10 個點只有一個會被收集與傳送。



注意

當數據點刻度太小時，或許會有過多的數據點且可能無法全部傳送。在這個例子中，部分的數據點在數據傳輸時會被捨棄。

數據點刻度參數只使用在連續模式的 DSP 示波器中，當在快照模式中，此參數會被忽略且每個點都會被收集與傳送。

在模擬中，SCI 輸出可視為電壓探測器。

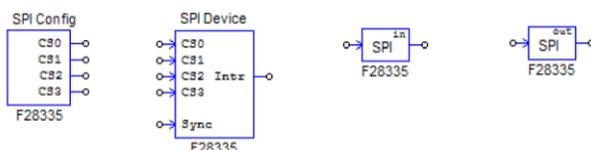
串列外設界面(SPI)

F2833x DSP 提供串列外設界面的功能。藉由 TI F2833x 標的資料庫中 SPI 方塊，可容易與方便地實現與外部 SPI 設備(如外部 A/D 與 D/A 轉換器)傳輸的功能。手寫 SPI 設備程式碼通常是一件耗時且不容易的任務。有了支援 SPI 功能，PSIM 大大地簡化並加速撰寫程式碼與硬體實現的過程。

對於如何使用 SPI 方塊更詳細的描述請參考“Tutorial - Using SCI for Real-time Monitoring in F2833x Target.pdf”文件。

SimCoder 提供四個 SPI 函數方塊：SPI 配置、SPI 設備、SPI 輸入與 SPI 輸出說明如下：

圖示



SPI 配置

SPI 配置方塊定義 SPI 端口、晶片選取引腳與 SPI 緩衝區大小。它必須存在於使用 SPI 的電路圖中且需位於主電路圖中。

屬性：

參數	描述
SPI Port	定義 SPI 端口，為 GPIO16-19 或 GPIO54-57。
Chip Select Pin0, 1, 2, and 3	晶片選擇引腳的 GPIO 端口。PSIM 最多支援 16 個 SPI 設備，其需要 4 個 GPIO 晶片選擇引腳，定義為晶片選擇引腳 1 到 3。GPIO 端口和 SPI slave transmit -enable(SPISTE)用來產生晶片選擇信號。

SPI Buffer Size SPI 命令的緩衝區大小。每個記憶體單元保存 SPI 命令的引索。通常可在所有 SPI 輸出入元件指定緩衝區大小為 1 加上 SPI 命令數(即開始轉換命令、接收數據命令、傳送數據命令與同步命令)。

SPI 設備

SPI 設備方塊定義相對應的 SPI 硬體設備的資訊。電路圖中 SPI 設備方塊的數量需與 SPI 硬體設備數相同。

屬性：

參數	描述
Chip Select Pins	SPI 設備對應的晶片選擇引腳之狀態。當晶片選擇引腳為此狀態時，則此 SPI 設備被選取。
Communication Speed(MHz)	SPI 傳輸速度，單位 MHz。
Clock Type	<p>SPI 時脈類型，由 SPI 硬體設備決定。可為下列其一：</p> <p>Rising edge without delay：時脈通常為低準位，且在上升緣時鎖住數據。</p> <p>Rising edge with delay：時脈通常為低準位，且在延遲的上升緣時鎖住數據。</p> <p>Falling edge without delay：時脈通常為高準位，且在下降緣時鎖住數據。</p> <p>Falling edge with delay：時脈通常為高準位，且在延遲的下降緣時鎖住數據。</p>
Command Word Length	SPI 傳輸命令的字長度或有效位元長度。可從 1 到 16 位元。
Sync. Active Mode	SPI 設備同步命令之觸發模式。可為上升緣或下降緣。

SPI Initial Command 初始化 SPI 設備的 SPI 命令。

Hardware Interrupt Mode 指定 SPI 設備產生的中斷信號類型。只有當 SPI 設備的中斷輸出節點連接到數位輸出元件的輸入時才有效。為下列其中之一：
No hardware interrupt
Rising edge
Falling edge

Interrupt Timing 指定當 SPI 設備完成轉換時如何產生中斷。可為下列其一：
No interrupt：無中斷產生。在這種情況，DSP 送命令給 SPI 輸入設備。此設備開始轉換並在相同命令下回傳結果。
Multiple interrupt in series：在每次轉換後產生多個串連中斷，針對有一個 A/D 轉換器與多個輸入通道的 SPI 設備。在這種情況中，DSP 送出第一個轉換命令且 SPI 設備開始轉換。當轉換完成，SPI 設備將產生中斷。在此中斷服務程序，DSP 將傳送一個命令以獲取轉換結果且相同 SPI 輸入設備的另個通道開始一個新的轉換。
One-time interrupt：在轉換結束時只產生一個中斷，SPI 設備可在一個請求中執行多個通道轉換。這種情況下，DSP 傳送命令到 SPI 輸入設備且 SPI 設備完成多個輸入通道的轉換。當轉換完成時，SPI 設備將產生中斷。

Command Gap(ns) 兩個 SPI 命令的間隔，單位奈秒(nsec)。

Conversion Sequence 定義 SPI 輸入元件的名稱，以逗號區隔並決定轉換順序。

 **注意** 只有當 SPI 設備產生多個串連中斷時變數才有效。

在電路圖中，所有 SPI 設備的晶片選取引腳連接到 SPI 配置方塊的晶片選取引腳，不須定義晶片選取邏輯如何實現。然而，在實際的硬體則需依據相對應的晶片選取邏輯實現。

SPI 命令為一連串藉由逗號隔開的 16 位元數字組成。在這 16 位元數字中，命令只使用較低位元為有效位元。例如，如果命令字長為 8，0~7 位元為命令，8~15 則未被使用。

SPI 設備可為輸出或輸入。例如，外部 A/D 轉換器為輸入設備，通常 DSP 會傳送一個或多個 A/D 轉換命令到此設備，然後設定一個同步信號開始轉換。同步信號在同一設備的下一個命令重置。

使用同步信號的 SPI 輸入設備通常需要一個中斷引腳去觸發 DSP 進入中斷服務程序。

另一方面，外部 D/A 轉換器為一輸出設備。通常 DSP 會傳送一個或多個 D/A 轉換命令到此設備，然後設定一個同步信號開始轉換。同步信號在同一設備的下一個命令重置。

SPI 輸入

SPI 輸入設備有多個輸入通道。SPI 輸入方塊用來定義 SPI 傳輸的輸入通道屬性且一個 SPI 輸入方塊對應一個輸入通道。

屬性：

參數	描述
Device name	SPI 輸入設備名稱
Start Conversion Command	開始轉換的命令，16 進制，以逗點區隔(例如，0x23, 0x43,0x00)。
Receiving Data Command	接收數據的命令，16 進制，以逗點區隔(例如，0x23, 0x43,0x00)。

Data Bit Position	<p>定義在接收的數據字串中數據位元的位置。公式為：$ElementName=\{Xn[MSB..LSB]\}$</p> <p>其中 $ElementName$ 為 SPI 輸入設備的名稱。如果是目前的 SPI 輸入設備，以 y 代替。</p> <p>$\{ \}$ 意味著括弧內項目重複多次。</p> <p>Xn 為從 SPI 輸入設備接收的第 n 個字，n 從 0 開始。</p> <p>$MSB..LSB$ 定義字的有效位元位置。</p>
Input Range	<p>指定參數 V_{max} 定義的輸入範圍。只有當 SPI 設備為 A/D 轉換器時此變數才有效。若設備轉換器模式為 DC，輸入範圍由 0 到 V_{max}。若設備轉換器模式為 AC，輸入範圍由 $-V_{max}/2$ 到 $V_{max}/2$。</p>
Scale Factor	<p>輸入比例因數 K_{scale}。若比例因數為 0，SPI 設備不為 A/D 轉換器，其結果將與 DSP 從 SPI 傳輸接收的完全一樣。反之，若 SPI 設備為 A/D 轉換器，其結果與該因數和 A/D 轉換模式成比例。</p>
ADC Mode	<p>設備的 A/D 轉換模式。可為 DC 或 AC，注意只有當此設備為 A/D 轉換器時參數才有效。</p>

Initial Value 輸入的初始值。

Data Bit Position 公式用來定義 SPI 輸入設備的數據長度。例如， $y=x1[3..0]x2[7..0]$ ，代表數據長度為 12，結果為第二個字的較低四個位元與第三個字的較低 8 個位元。如果接收到的字串為 0x12,0x78,0xAF，結果為 0x8AF。

如果比例因數不為 0，則輸出將與下列成比例：

- 在 DC 轉換模式
- 模擬： $Output = Input \cdot K_{scale}$
 - 硬體： $Output = \frac{Result \cdot V_{max} \cdot K_{scale}}{2^{Data_Length}}$

- 在 AC 轉換模式
- 模擬： $Output = Input \cdot K_{scale}$
 - 硬體： $Output = \frac{(Result - 2^{Data_Length-1}) \cdot V_{max}}{2^{Data_Length-1}} \cdot K_{scale}$

$Data_Length$ 參數由 Data Bit Position Formula 計算得之。

SPI 輸出

SPI 輸出設備有多個 SPI 輸出通道。SPI 輸出方塊用來定義 SPI 傳輸的輸出通道屬性且一個 SPI 輸出方塊對應一個輸出通道。

屬性

參數	描述
Device name	SPI 輸出設備名稱
Scale Factor	輸出比例因數 Kscale。若比例因數為 0，SPI 設備不為 A/D 轉換器，其結果將與 DSP 從 SPI 傳輸接收的完全一樣。反之，若 SPI 設備為 A/D 轉換器，其結果與該因數和 A/D 轉換模式成比例。
Output Range	指定參數 Vmax 定義的輸出範圍。只有當 SPI 設備為 A/D 轉換器時此變數才有效。若設備轉換器模式為 DC，輸入範圍由 0 到 Vmax。若設備轉換器模式為 AC，輸入範圍由 -Vmax/2 到 Vmax/2。
DAC Mode	設備的 D/A 轉換模式。可為 DC 或 AC，注意只有當此設備為 D/A 轉換器時參數才有效。
Sending Data Command	傳送輸出數據的命令，16 進制，以逗點區隔(例如，0x23,0x43,0x00)。
Data Bit Position	定義在傳送的數據字串中數據位元的位置。公式為： $ElementName=\{Xn[MSB..LSB]\}$ 其中 ElementName 為 SPI 輸出設備的名稱。如果是目前的 SPI 輸出設備，以 y 代替。 { } 意味著括弧內項目重複多次。 Xn 為從 SPI 輸出設備傳送的第 n 個字，n 從 0 開始。 MSB..LSB 定義字的有效位元位置。

SPI 輸出設備的同步輸出通道之命令，16 進制，以
 Sync. Command 逗點區隔(例如，0x23,0x43,0x00)。當 SPI 輸出設備
 無同步信號時才使用此命令。

Data Bit Position 公式用來定義 SPI 輸出設備的數據長度。例如，
 $y=x1[3..0]x2[7..0]$ 代表數據長度為 12，結果為第二個字的較低四個位
 元與第三個字的較低 8 個位元。如果接收到的字串為
 0x12,0x78,0xAF，結果為 0x8AF。

如果比例因數不為 0，則輸出將與下列成比例：

在 DC 轉換模式 • 模擬： $Output = Input \cdot K_{scale}$

$$Output = \frac{Result \cdot K_{scale} \cdot 2^{Data_Length}}{V_{max}}$$

• 硬體：

在 AC 轉換模式 • 模擬： $Output = Input \cdot K_{scale}$

$$Output = 2^{Data_Length} + \frac{Result \cdot K_{scale} \cdot 2^{Data_Length-1}}{V_{max}}$$

Data_Length 參數由 Data Bit Position Formula 計算得之。

專案設定與記憶體配置

當針對 F2833x 硬體標的產生程式碼時，SimCoder 也對於程式碼編譯、連結與上傳至 DSP 開發環境 TI CCS 創造一個完整的專案文件。

目前，支援 CCS 3.3 版。假定 PSIM 電路圖文件是“test.sch”，在產生程式碼後，電路圖文件的目錄下將產生“test(C Code)”的子資料夾並包含下列文件：

- test.c：產生的 C 程式碼
- PS_bios.h：SimCoder F2833x 資料庫的標題文件
- passwords.asm：說明 DSP 程式碼密碼的文件
- test.pjt：Code Composer Studio 的專案文件
- DSP28335_Headers_nonBIOS.cmd：外設暫存器連接器命令文件
- F28335_FLASH_Lnk.cmd：Flash memory 快閃記憶體連接器命令文件
- F28335_FLASH_RAM_Lnk.cmd：Flash RAM memory 連接器命令文件
- F28335_RAM_Lnk.cmd：RAM memory 連接器命令文件



注意

如果硬體標的不是 F28335，連接器命令文件的名稱將被指定給對應的標的硬體。例如，如果標的硬體為 F28334，文件名稱相對地為 F28334 FLASH Lnk.cmd, F28334 FLASH RAM Lnk.cmd 與 F28334RAM Lnk.cmd。

除此之外，專案也需要下列文件：

- PS_bios.lib：SimCoder F2833x 資料庫，位於 PSIM 資料夾
- C28x_FPU_FastRTS_beta1.lib：TI 快速浮點資料庫，位於 PSIM\lib 子資料夾

當程式碼產生時，會自動複製"C28x_FPU_FastRTS_beta1.lib"和"PS_bios.lib"這兩個文件到專案資料夾。

每次產生程式碼時，也會創造.c與.pjt文件(此範例中為test.c和test.pjt)。如果已手動更改這兩個文件，一定要複製更改的文件到不同的位置。否則，下次執行產生程式碼時，更改的文件將會被覆蓋。

(17) 專案設定

在 CCS 專案文件，需提供下列設定：

- *RAM Debug*：在 Debug 模式編譯程式碼並 RAM memory 在執行。
- *RAM Release*：在 Release 模式編譯程式碼並 RAM memory 在執行。
- *Flash Release*：在 Release 模式編譯程式碼並 Flash memory 在執行。
- *Flash RAM Release*：在 Release 模式編譯程式碼並 RAM memory 在執行。

當選擇 RAM Debug 或 RAM Release 設定時，CCS 使用連接器命令文件 F28335_RAM_Lnk.cmd 配置程式與數據空間。

當選擇 Flash Release 設定時，CCS 使用連接器命令文件 F28335_FLASH_Lnk.cmd 配置程式與數據空間。

當選擇 Flash RAM Release 設定時，CCS 使用連接器命令文件 F28335_FLASH_RAM_Lnk.cmd 配置程式與數據空間。記憶體配置相同於 RAM Release 設定。

在 release 模式下編輯程式碼快過在 debug 模式，在 RAM Release 和 Flash RAM Release 最快，RAM Debug 則較慢，Flash Release 則是最慢。在開發中，通常會先在 RAM Debug 以易於 debug，然後切換到 RAM Release 隨後到 RAM Release 或 Flash RAM Release。

(18) 記憶體配置

在產生連結的文件，記憶體配置由下列方式定義：

RAM Debug、RAM Release 與 Flash RAM Release 的設置：

RAM Memory
0x0000 - 0x07FF (2K)
interrupt vectors
stack
0x8000 - 0xFFFF (32K*)
program and data space

Flash Release 的設置

RAM Memory
0x0000 - 0x07FF (2K)
interrupt vectors
stack
0x8000 - 0xFFFF (32K*)
data space

Flash Memory
0x300000 - 0x33FFFF (256K**)
program
password
etc.



1. SimCoder 預先定義了用於程式與數據空間的 RAM memory：
 - 針對 F28335, F28334：從 0x8000 到 0xFFFF (32K)
 - 針對 F28332：從 0x8000 到 0xDFFF (16K)
 - 如果程序加上數據空間超過 RAM 空間的大小，則必須選擇 Flash Release 作為專案設定。
2. SimCoder 預先定義了用於程式空間的 flash memory：
 - 針對 F28335：從 0x300000 到 0x33FFFF (256K)
 - 針對 F28334：從 0x320000 到 0x33FFFF (128K)
 - 針對 F28332：從 0x330000 到 0x33FFFF (64K)