SyncMOS Technologies International, Inc.

ADC
Technical
Application Notes

Index

1. 簡述

本應用指南提供了該系列產品的一些功能應用及所需要注意的訊息或問題的解決對策，以作為改善客戶所碰到的疑難問題，不過相關的功能應用及電器特性等，客戶還是需要請參考 Datasheet. 可於新茂網站 www.syncmos.com.tw 下載。

2. 適用產品

零 件 號 碼
SM59A16U1、SM59R04A2、SM59R05A5、SM59R09A5、SM59R16A5、SM39A16M1、 SM39R16A6、SM39R08A2、SM39R16A2、SM39R08A3、SM39R16A3、SM39R08A5

3. 應用說明

3.1 ADC規格概述

(1). 10-bit 的 ADC 可提供最大的通道數及轉換率如表 3-1。

* 請看 Keil C 有關 C51 使用者指南中的描述中斷功能。

表 3-1: ADC 通道數及轉換率

通道數	ADC Clock 最大頻率(MHz)	最高轉換率 (KHz)	零 件 號 碼
9	12.5	961	SM59A16U1
8	12.5	500	SM59R04A2、 SM59R05A5、SM59R09A5、SM59R16A5
8	12.5	961	SM39A16M1、SM39R16A6
8	12.5	500	SM39R08A2、SM39R16A2
7	12.5	961	SM39R08A3、SM39R16A3
8	11.0592	851	SM39R08A5

(2). ADC 為 SAR 架構。

(3). ADC Clock，提供 32 組預除設定（請參考 SFR ADCCS 設定）。

(4). ADC 中斷向量為 0x53H，中斷號碼為 10，如表 3-2。

表 3-2: ADC 相對應的中斷向量表

中斷要求旗標	中斷向量位址	中斷向量號碼 *(use Keil C Tool)
IE0 – External interrupt 0	0003h	0
TF0 – Timer 0 interrupt	000Bh	1
IE1 – External interrupt 1	0013h	2
.	.	.
.	.	.
.	.	.
.	.	.
ADCIF – A/D converter interrupt	**0053h**	**10**
.	.	.
.	.	.
IICIF – IIC interrupt	006Bh	13
RI1/TI1 – Serial channel 1 interrupt	0083h	16

*See Keil C about C51 User's Guide about Interrupt Function 描述

3.2 SM59A16U1

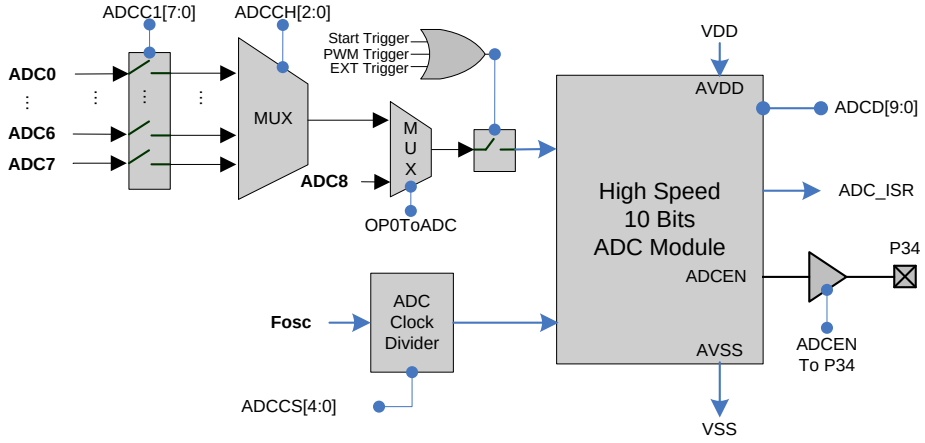


圖 3-1: ADC 模組工作方塊圖

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
ADC											
ADCC1	ADC Control register 1	ABh	ADC7 EN	ADC6 EN	ADC5E N	ADC4 EN	ADC3 EN	ADC2 EN	ADC1 EN	ADC0 EN	00H
ADCC2	ADC Control register 2	ACh	Start	ADJU ST	PWM Trigger EN	EXT Trigge rEN	ADC MODE	ADCCH[2:0]			00H
ADCDH	ADC data high byte	ADh	ADCDH [7:0]								00H
ADCDL	ADC data low byte	AEh	ADCDL [7:0]								00H
ADCCS	ADC clock select	AFh	OP0 ToAD C	-	ADCEN ToP34	ADCCS[4:0]					00H
IEN1	Interrupt Enable 1 register	B8h	EXEN 2	-	IEIIC	IELVI	IEKBI	IEAD C	IESPI	-	00h
IRCON	Interrupt request register	C0H	EXF2	TF2	IICIF	LVIIF	KBIIF	ADCIF	SPIIF	PWMIF	00H

Mnemonic: ADCC1 **Address: ABh**

7	6	5	4	3	2	1	0	Reset
ADC7EN	ADC6EN	ADC5EN	ADC4EN	ADC3EN	ADC2EN	ADC1EN	ADC0EN	00H

ADC7EN: 致能 ADC 通道 7.

ADC7EN = 1 –致能 ADC 通道 7

ADC6EN: 致能 ADC 通道 6.

ADC6EN = 1 –致能 ADC 通道 6

ADC5EN: 致能 ADC 通道 5.

ADC5EN = 1 –致能 ADC 通道 5

ADC4EN: 致能 ADC 通道 4.

ADC4EN = 1 –致能 ADC 通道 4

ADC3EN: 致能 ADC 通道 3.

ADC3EN = 1 –致能 ADC 通道 3

ADC2EN: 致能 ADC 通道 2.

ADC2EN = 1 –致能 ADC 通道 2

ADC1EN: 致能 ADC 通道 1.

ADC1EN = 1 –致能 ADC 通道 1

ADC0EN: 致能 ADC 通道 0.

ADC0EN = 1 –致能 ADC 通道 0

Mnemonic: ADCC2 **Address: ACh**

7	6	5	4	3	2 1 0	Reset
Start	ADJUST	PWMTriggerEN	EXTTriggerEN	ADCMODE	ADCCH[2:0]	00H

Start: 當該位元被置位時，ADC 將啟動轉換.

ADJUST: ADC 數位輸出格式調整.

ADJUST = 0: (初始值)

ADC 數位元輸出高位元組 ADCD [9:2] = ADCDH [7:0].

ADC 數位輸出低位元組 ADCD [1:0] = ADCDL [1:0].

ADJUST = 1:

ADC 數位元輸出高位元組 ADCD [9:8] = ADCDH [1:0].

ADC 數位輸出低位元組 ADCD [7:0] = ADCDL [7:0].

PWMTriggerEN PWM 觸發 ADC 開始轉換

（HW 內部觸發轉換）

0 =禁能

1 =致能

EXTTriggerEN 外部 Pin 腳觸發 ADC 開始轉換

（HW 外部觸發轉換）

0 =禁能

1 =致能

ADCMOD 0 =連續模式

1 =單次轉換模式

ADCCH[2:0]: ADC 通道選擇.

ADCCH [2:0]	通道
000	0
001	1
010	2
011	3
100	4
101	5

110	6
111	7

ADJUST = 0:

Mnemonic: ADCDH **Address: ADh**

7	6	5	4	3	2	1	0	Reset
ADCD[9]	ADCD[8]	ADCD[7]	ADCD[6]	ADCD[5]	ADCD[4]	ADCD[3]	ADCD[2]	00H

Mnemonic: ADCDL **Address: AEh**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	ADCD[1]	ADCD[0]	00H

ADJUST = 1:

Mnemonic: ADCDH **Address: ADh**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	ADCD[9]	ADCD[8]	00H

Mnemonic: ADCDL **Address: AEh**

7	6	5	4	3	2	1	0	Reset
ADCD[7]	ADCD[6]	ADCD[5]	ADCD[4]	ADCD[3]	ADCD[2]	ADCD[1]	ADCD[0]	00H

ADCD[9:0]: ADC 數字暫存器.

Mnemonic: ADCCS **Address: AFh**

7	6	5	4	3	2	1	0	Reset
OP0ToADC	-	ADCENToP34	ADCCS[4]	ADCCS[3]	ADCCS[2]	ADCCS[1]	ADCCS[0]	00H

OP0ToADC: 選擇 ADC 通道 8 作為輸入源

0 = 設置 ADC 輸入源由 ADCC2 決定.

1 = 設置 ADC 輸入源作為 Op0 輸出.

ADCENToP34: ADC 內部信號測試和監視器.

0 = 禁能 ADC 內部信號輸出至 P3.4

1 = 致能 ADC 內部信號輸出至 P3.4

ADCCS[4:0]: ADC Clock 選擇.

* ADC Clock 最大為 12.5MHz.

* ADC 轉換率最高為 961KHz

$$ADC\_Clock = \frac{\text{Fosc}}{2 \times (ADCCS + 1)}$$

$$ADC\_Conversion\_Rate = \frac{\text{ADC\_Clock}}{13}$$

Mnemonic: IEN1 **Address: B8h**

7	6	5	4	3	2	1	0	Reset
EXEN2	-	IEIIC	IELVI	IEKBI	IEADC	IESPI	IEPWM	00h

IEADC: A/D 轉換中斷致能位元元

IEADC = 0 –致能 ADC 中斷.

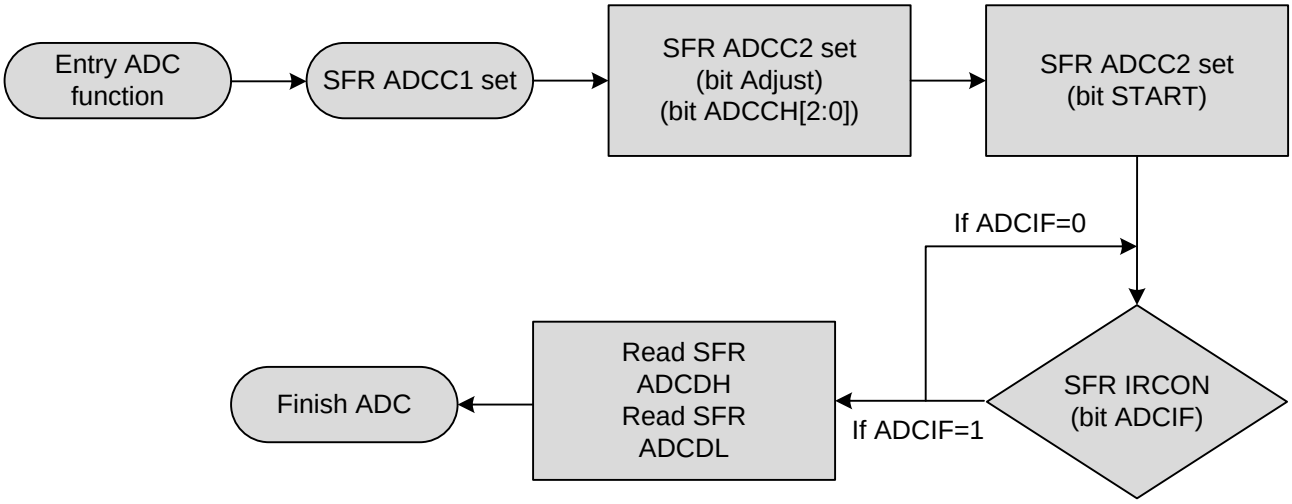
IEADC = 1 –致能 ADC 中斷.

Mnemonic: IRCON **Address: C0h**

7	6	5	4	3	2	1	0	Reset
EXF2	TF2	IICIF	LVIIF	KBIIF	ADCIF	SPIIF	PWMIF	00H

ADCIF: A/D 轉換中斷旗標位元元，當有開啟 ADC 中斷時，轉換完成會設為 1，進中斷後硬體自動清為 0.若未開 ADC 中斷，則必須手動軟體清除為 0.

3.2.1 ADC一般應用流程圖

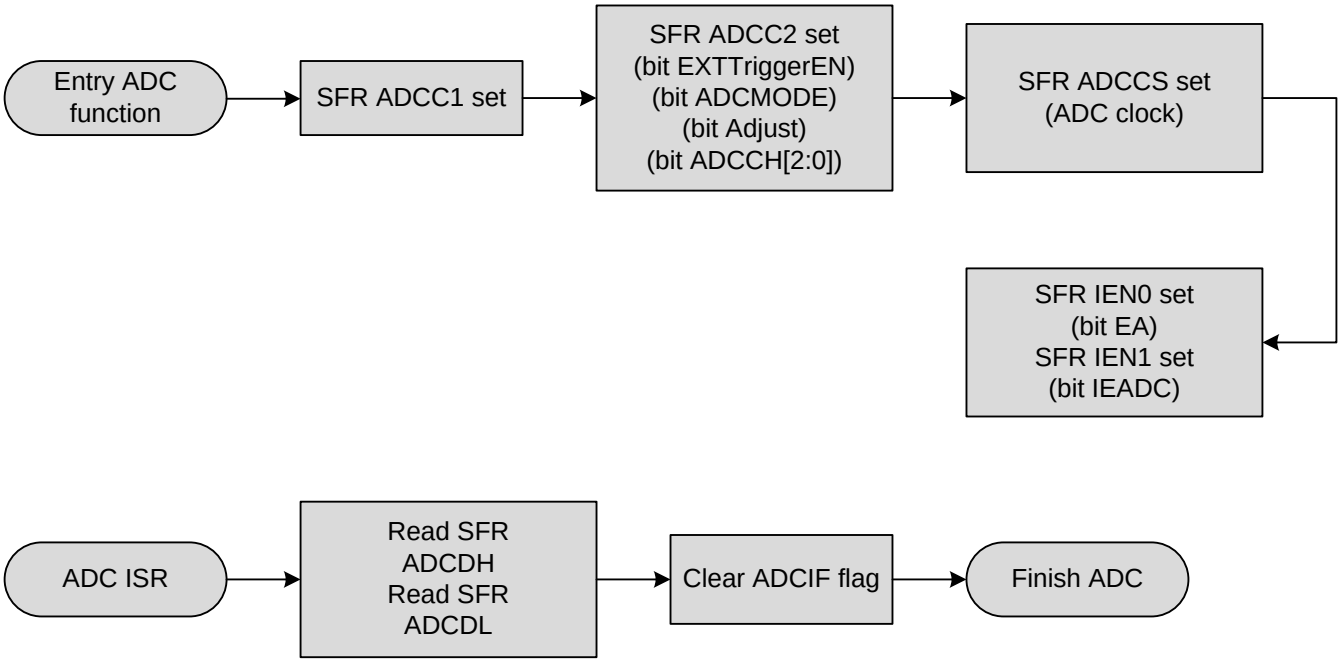


3.2.2 ADC一般程式範例

Describe:	Program:

main	//==========
	//
	// SYNCMOS TECHNOLOGY
	//
	//==========
	#include "..\h\SM39A16U1.h"
	
	void main(void)
	{
	unsigned char temp_H,temp_L;
	
	ADCC1 = 0x01; //ADC Chanel 0 enable
	ADCC2 = 0x00; //Continuous mode, ADC Chanel 0 is analog input, Adiust=0
	ADCCS = 0x00; //ADC clock Fosc/2
	
	while(1)
	{
	ADCC2 |= 0x80; //sbit ADC START = 1, will auto clear after finish
	while(!IRCON && 0x04); //finish if ADCIF=1, converting if ADCIF=0
	temp_L = ADCDL; //ADC result, the Adjust=0, the ADCDL[1:0]=ADCD[1:0]
	temp_H = ADCDH; //the ADCDH[7:0]=ADCD[9:2]
	IRCON &= 0xFB; //Clear ADCIF flag for next ADC conversion
	}
	}

3.2.3 ADC外部觸發應用流程圖

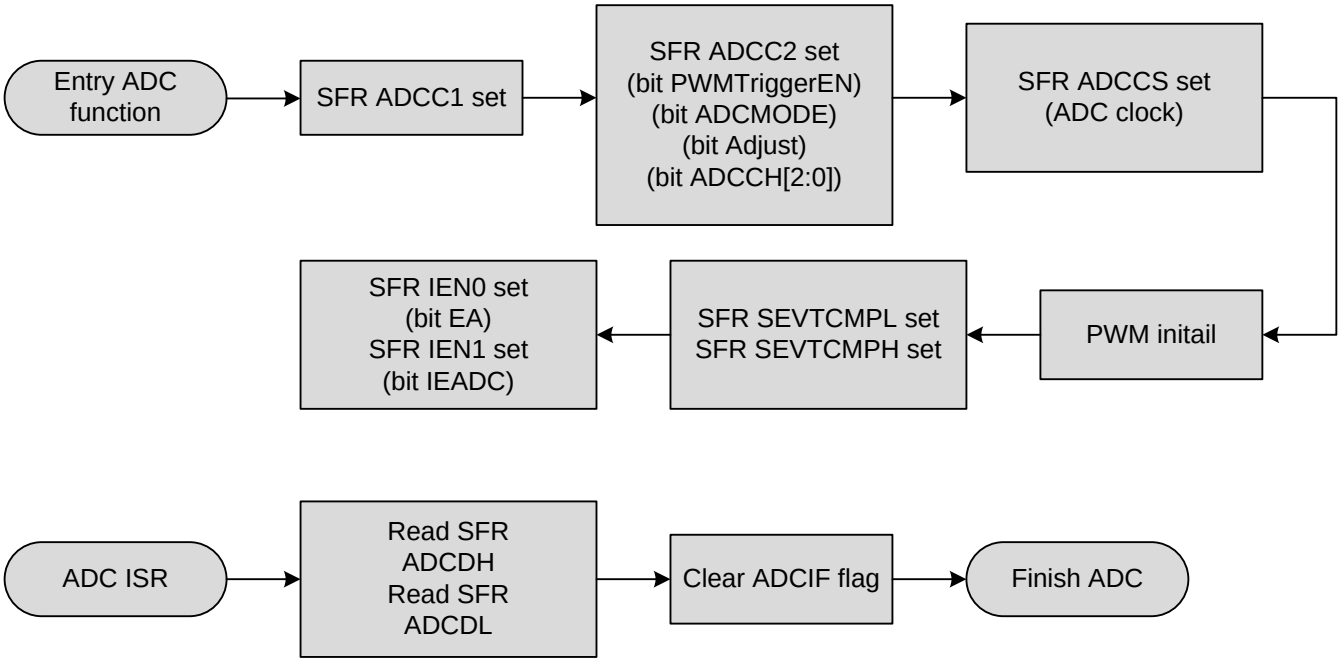


3.2.4 ADC外部觸發程式範例

Describe:	Program:

main	//==========
	//
	// SYNCMOS TECHNOLOGY
	//
	//==========
	#include "..\h\SM39A16U1.h"
	
	void main(void)
	{
	
	ADCC1 = 0x01; //ADC Chanel 0 enable
	ADCC2 = 0x18; //EXT TriggerADC Enable, single-shot mode, ADC Chane 0 is
	// analog input, Adiust=0
	ADCCS = 0x00; //ADC clock Fosc/2
	
	EA = 1; //Enable all inierrupt
	IEADC = 1; //Enable ADC inierrupt
	while(1){}
	}
	
	void ADC_ISR(void) interrupt d_ADC_Vector // Address: 0x53
	{
	unsigned char temp_H,temp_L;
	
	temp_L = ADCDL; //ADC result, the Adjust=0, the ADCDL[1:0]=ADCD[1:0]
	temp_H = ADCDH; //the ADCDH[7:0]=ADCD[9:2]
	ADCIF =0; //Clear ADCIF flag for next ADC conversion
	}

3.2.5 PWM觸發ADC應用流程圖

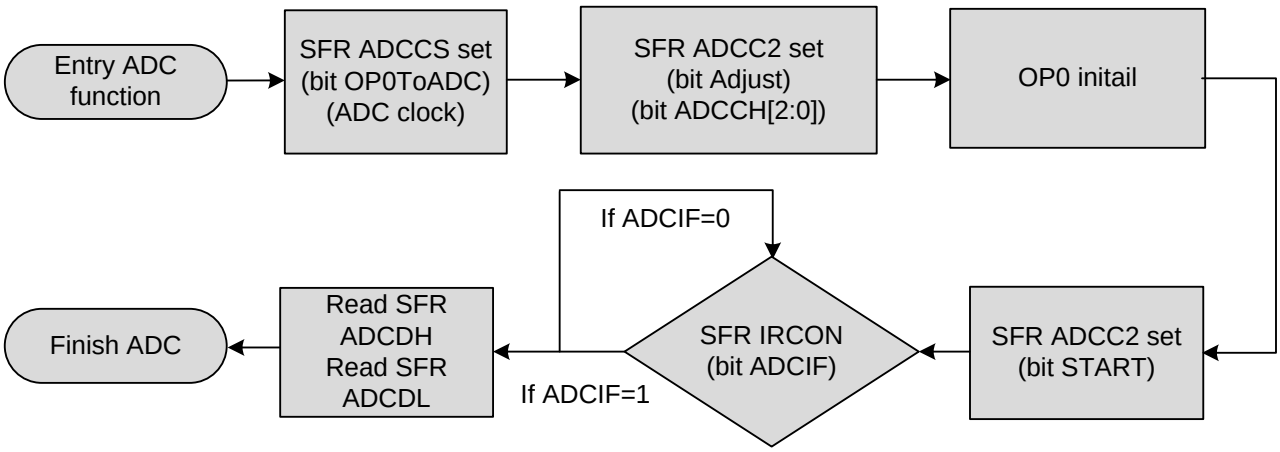


3.2.6 PWM觸發ADC程式範例

Describe:	Program:

main	//==========
	//
	// SYNCMOS TECHNOLOGY
	//
	//==========
	#include "..\h\SM39A16U1.h"
	
	void main(void)
	{
	
	PAGESEL = 0x01; //Page Mode: Page0
	ADCC1 = 0x01; //ADC Chanel 0 enable
	ADCC2 = 0x28; //PWM TriggerADC Enable, single-shot mode, ADC Chanel
	// 0 is analog input, Adiust=0
	ADCCS = 0x00; //ADC clock Fosc/2
	PAGESEL = 0x03; //Page Mode: Page1
	SEVTCMPL = 0x00; //Special Event Compare Low byte
	SEVTCMPH = 0x00; //Special Event Compare High byte
	PAGESEL = 0x01; //Page Mode: Page0
	PWM_Init();
	EA = 1; //Enable all inierrupt
	IEADC = 1; //Enable ADC inierrupt
	while(1){}
	}
	
	void ADC_ISR(void) interrupt d_ADC_Vector // Address: 0x53
	{
	unsigned char temp_H,temp_L;
	
	temp_L = ADCDL; //ADC result, the Adjust=0, the ADCDL[1:0]=ADCD[1:0]
	temp_H = ADCDH; //the ADCDH[7:0]=ADCD[9:2]
	ADCIF =0; //Clear ADCIF flag for next ADC conversion
	}

3.2.7 OP0輸出轉ADC應用流程圖



3.2.8 OP0輸出轉ADC程式範例

Describe:	Program:

main	//===
	//
	// SYNCMOS TECHNOLOGY
	//
	//===
	#include "..\h\SM39A16U1.h"
	
	void main(void)
	{
	unsigned char temp_H,temp_L;
	
	ADCCS = 0x80; //ADC Chanel 8 enable, ADC clock Fosc/2
	ADCC2 = 0x00; //Continuous mode, Adiust=0
	OP0_Init();
	
	while(1)
	{
	ADCC2 |= 0x80; //sbit ADC START = 1, will auto clear after finish
	while(!IRCON && 0x04); //finish if ADCIF=1, converting if ADCIF=0
	temp_L = ADCDL; //ADC result, the Adjust=0, the ADCDL[1:0]=ADCD[1:0]
	temp_H = ADCDH; //the ADCDH[7:0]=ADCD[9:2]
	IRCON &= 0xFB; //Clear ADCIF flag for next ADC conversion
	}
	}

3.3 SM59R04A2、SM59R05A5、SM59R09A5、SM59R16A5

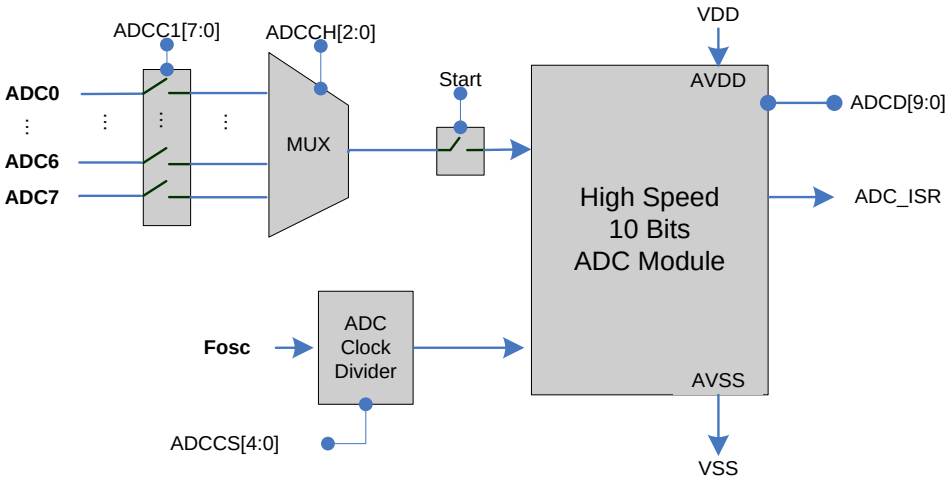


圖 3-2: ADC 模組工作方塊圖

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
ADC											
ADCC1	ADC Control register 1	ABh	ADC7 EN	ADC6 EN	ADC5E N	ADC4 EN	ADC3 EN	ADC2 EN	ADC1 EN	ADC0 EN	00H
ADCC2	ADC Control register 2	ACh	Start	ADJUST	-	-	-	ADCCH[2:0]			00H
ADCDH	ADC data high byte	ADh	ADCDH [7:0]								00H
ADCDL	ADC data low byte	AEh	ADCDL [7:0]								00H
ADCCS	ADC clock select	AFh	-	-	-	ADCCS[4:0]					00H
IEN1	Interrupt Enable 1 register	B8h	EXEN2	-	IEIIC	IELVI	IEKBI	IEADC	IESPI	IEPWM	00H
IRCON	Interrupt request register	C0H	EXF2	TF2	IICIF	LVIIF	KBIIF	ADCIF	SPIIF	PWMIF	00H

Mnemonic: ADCC1 **Address: ABh**

7	6	5	4	3	2	1	0	Reset
ADC7EN	ADC6EN	ADC5EN	ADC4EN	ADC3EN	ADC2EN	ADC1EN	ADC0EN	00H

ADC7EN: 致能 ADC 通道 7.

ADC7EN = 1 –致能 ADC 通道 7

ADC6EN: 致能 ADC 通道 6.

ADC6EN = 1 –致能 ADC 通道 6

ADC5EN: 致能 ADC 通道 5.

ADC5EN = 1 –致能 ADC 通道 5

ADC4EN: 致能 ADC 通道 4.

ADC4EN = 1 –致能 ADC 通道 4

ADC3EN: 致能 ADC 通道 3.

ADC3EN = 1 –致能 ADC 通道 3

ADC2EN: 致能 ADC 通道 2.

ADC2EN = 1 –致能 ADC 通道 2

ADC1EN: 致能 ADC 通道 1.

ADC1EN = 1 –致能 ADC 通道 1

ADC0EN: 致能 ADC 通道 0.

ADC0EN = 1 –致能 ADC 通道 0

Mnemonic: ADCC2 **Address: ACh**

7	6	5	4	3	2	1	0	Reset
Start	ADJUST	-	-	-	ADCCH[2:0]			00H

Start: 當該位元元被置位時，ADC 將啟動轉換.

ADJUST: DC 數位輸出格式調整.

ADJUST = 0: (初始值)

ADC 數位元輸出高位元組 ADCD [9:2] = ADCDH [7:0].

ADC 數位輸出低位元組 ADCD [1:0] = ADCDL [1:0]..

ADJUST = 1:

ADC 數位元輸出高位元組 ADCD [9:8] = ADCDH [1:0].

ADC 數位輸出低位元組 ADCD [7:0] = ADCDL [7:0].

ADCCH[2:0]: ADC 通道選擇.

ADCCH [2:0]	Channel
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

ADJUST = 0:

Mnemonic: ADCDH **Address: ADh**

7	6	5	4	3	2	1	0	Reset
ADCD[9]	ADCD[8]	ADCD[7]	ADCD[6]	ADCD[5]	ADCD[4]	ADCD[3]	ADCD[2]	00H

Mnemonic: ADCDL **Address: AEh**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	ADCD[1]	ADCD[0]	00H

ADJUST = 1:

Mnemonic: ADCDH **Address: ADh**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	ADCD[9]	ADCD[8]	00H

Mnemonic: ADCDL **Address: AEh**

7	6	5	4	3	2	1	0	Reset
ADCD[7]	ADCD[6]	ADCD[5]	ADCD[4]	ADCD[3]	ADCD[2]	ADCD[1]	ADCD[0]	00H

ADCD[9:0]: ADC 數字暫存器.

Mnemonic: ADCCS **Address: AFh**

7	6	5	4	3	2	1	0	Reset
-	-	-	ADCCS[4]	ADCCS[3]	ADCCS[2]	ADCCS[1]	ADCCS[0]	00H

ADCCS[4:0]: ADC Clock 選擇.

* ADC Clock 最大為 12.5MHz.
* ADC 轉換率最高為 500KHz

$$ADC\_Clock = \frac{Fosc}{2 \times (ADCCS + 1)}$$

$$ADC\_Conversion\_Rate = \frac{ADC\_Clock}{23}$$

Mnemonic: IEN1 **Address: B8h**

7	6	5	4	3	2	1	0	Reset
EXEN2	-	IEIIC	IELVI	IEKBI	IEADC	IESPI	IEPWM	00h

IEADC: A/D 轉換中斷致能位元元

IEADC = 0 –致能 ADC 中斷.

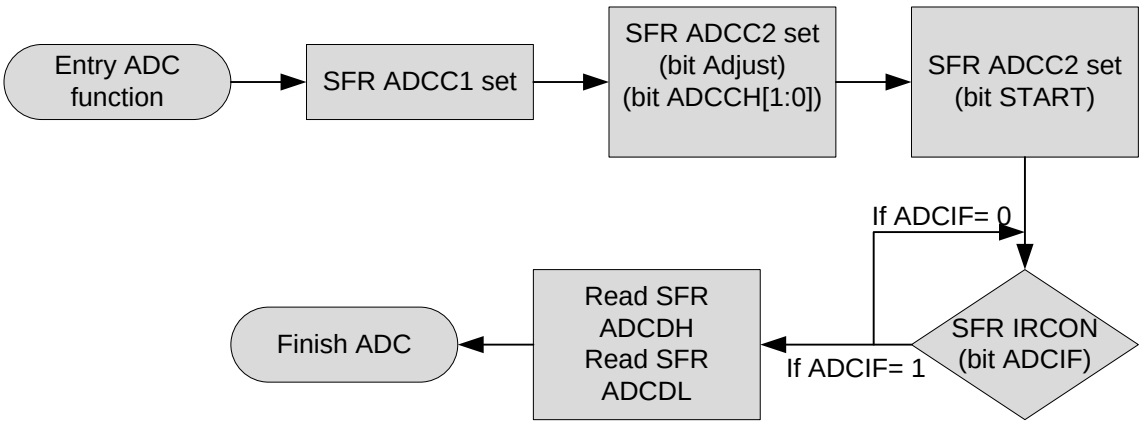
IEADC = 1 –致能 ADC 中斷.

Mnemonic: IRCON **Address: C0h**

7	6	5	4	3	2	1	0	Reset
EXF2	TF2	IICIF	LVIIF	KBIIF	ADCIF	SPIIF	PWMIF	00H

ADCIF: A/D 轉換中斷旗標位元，當有開啟 ADC 中斷時，轉換完成會設為 1，進中斷後硬體自動清為 0.若未開 ADC 中斷，則必須手動軟體清除為 0.

3.3.1 ADC應用流程圖



3.3.2 ADC程式範例

Describe:	Program:

main	//===
	//
	// S Y N C M O S T E C H N O L O G Y
	//
	//===
	#include "..\h\SM59R04A2.h"
	
	void main(void)
	{
	unsigned char temp_H,temp_L;
	
	ADCC1 = 0x01; //ADC Chanel 0 enable
	ADCC2 = 0x00; //ADC Chanel 0 is analog input, Adiust=0
	
	while(1)
	{
	ADCC2 |=0x80; //sbit ADC START = 1, will auto clear after finish
	while(!IRCON && 0x04); //finish if ADCIF=1, converting if ADCIF=0
	
	temp_L = ADCDL;/ /ADC result, the Adjust=0, the ADCDL[1:0]=ADCD[1:0]
	temp_H = ADCDH; //the ADCDH[7:0]=ADCD[9:2]
	IRCON &= 0xFB; //Clear ADCIF flag for next ADC conversion
	}
	}

3.4 SM39A16M1、SM39R16A6

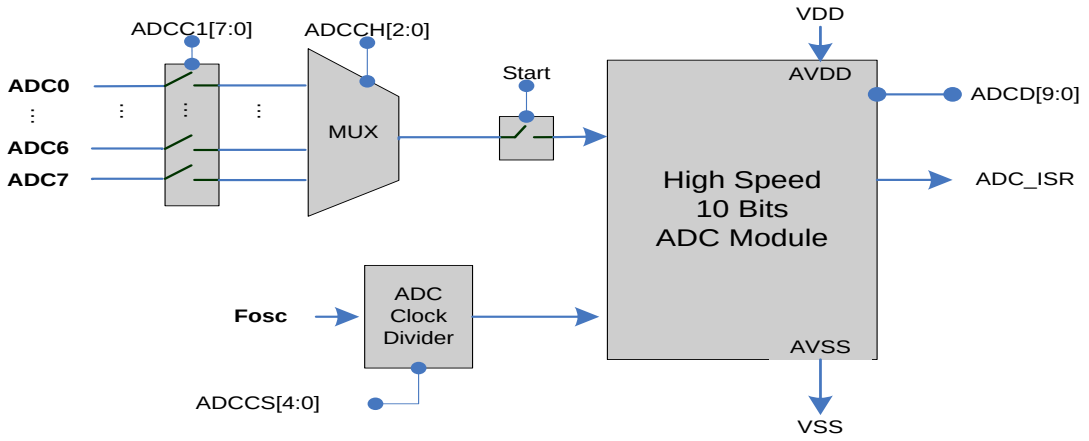


圖 3-3: ADC 模組工作方塊圖

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
ADC											
ADCC1	ADC Control register 1	ABh	ADC7 EN	ADC6 EN	ADC5 EN	ADC4 EN	ADC3 EN	ADC2 EN	ADC1 EN	ADC0 EN	00H
ADCC2	ADC Control register 2	ACh	Start	ADJU ST	PWMT rigger EN	EXTTri ggerE N	ADCM ODE	ADCCH[2:0]			08H
ADCDH	ADC data high byte	ADh	ADCDH [7:0]								00H
ADCDL	ADC data low byte	AEh	ADCDL [7:0]								00H
ADCCS	ADC clock select	AFh	-	-	-	ADCCS[4:0]					00H
IEN1	Interrupt Enable 1 register	B8h	EXEN 2	-	IEIIC	IELVI	-	IEAD C	IESPI	-	00H
IRCON	Interrupt request register	C0H	EXF2	TF2	IICIF	LVIIF	-	ADCIF	SPIIF	-	00H

Mnemonic: ADCC1 **Address: ABh**

7	6	5	4	3	2	1	0	Reset
ADC7EN	ADC6EN	ADC5EN	ADC4EN	ADC3EN	ADC2EN	ADC1EN	ADC0EN	00H

ADC7EN: 致能 ADC 通道 7.

ADC7EN = 1 –致能 ADC 通道 7

ADC6EN: 致能 ADC 通道 6.

ADC6EN = 1 –致能 ADC 通道 6

ADC5EN: 致能 ADC 通道 5.

ADC5EN = 1 –致能 ADC 通道 5

ADC4EN: 致能 ADC 通道 4.

ADC4EN = 1 –致能 ADC 通道 4

ADC3EN: 致能 ADC 通道 3.

ADC3EN = 1 –致能 ADC 通道 3

ADC2EN: 致能 ADC 通道 2.

ADC2EN = 1 –致能 ADC 通道 2

ADC1EN: 致能 ADC 通道 1.

ADC1EN = 1 –致能 ADC 通道 1

ADC0EN: 致能 ADC 通道 0.

ADC0EN = 1 –致能 ADC 通道 0

Mnemonic: ADCC2 **Address: ACh**

7	6	5	4	3	2 1 0	Reset
Start	ADJUST	PWMTriggerEN	EXTTriggerEN	ADCMODE	ADCCH[2:0]	08H

Start: 當該位元被置位時，ADC 將啟動轉換.

ADJUST: ADC 數位輸出格式調整.

ADJUST = 0: (初始值)

ADC 數位元輸出高位元組 ADCD [9:2] = ADCDH [7:0].

ADC 數位輸出低位元組 ADCD [1:0] = ADCDL [1:0].

ADJUST = 1:

ADC 數位元輸出高位元組 ADCD [9:8] = ADCDH [1:0].

ADC 數位輸出低位元組 ADCD [7:0] = ADCDL [7:0].

PWMTriggerEN PWM 觸發 ADC 開始轉換

（HW 內部觸發轉換）

0 =禁用

1 =致能

EXTTriggerEN 外部 Pin 腳觸發 ADC 開始轉換

（HW 外部觸發轉換）

0 =禁用

1 =致能

ADCMODE 0 =連續模式

1 =單次轉換模式

ADCCH[2:0]: ADC 通道選擇.

ADCCH [2:0]	通道
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

ADJUST = 0:

Mnemonic: ADCDH **Address: ADh**

7	6	5	4	3	2	1	0	Reset
ADCD[9]	ADCD[8]	ADCD[7]	ADCD[6]	ADCD[5]	ADCD[4]	ADCD[3]	ADCD[2]	00H

Mnemonic: ADCDL **Address: AEh**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	ADCD[1]	ADCD[0]	00H

ADJUST = 1:

Mnemonic: ADCDH **Address: ADh**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	ADCD[9]	ADCD[8]	00H

Mnemonic: ADCDL **Address: AEh**

7	6	5	4	3	2	1	0	Reset
ADCD[7]	ADCD[6]	ADCD[5]	ADCD[4]	ADCD[3]	ADCD[2]	ADCD[1]	ADCD[0]	00H

ADCD[9:0]: ADC 數字暫存器.

Mnemonic: ADCCS **Address: AFh**

7	6	5	4	3	2	1	0	Reset
-	-	-	ADCCS[4]	ADCCS[3]	ADCCS[2]	ADCCS[1]	ADCCS[0]	00H

ADCCS[4:0]: ADC Clock 選擇.

* ADC Clock 最大為 12.5MHz.
* ADC 轉換率最高為 961KHz

$$ADC\_Clock = \frac{\text{Fosc}}{2\times(ADCCS+1)}$$

$$ADC\_Conversion\_Rate = \frac{\text{ADC\_Clock}}{13}$$

Mnemonic: IEN1 **Address: B8h**

7	6	5	4	3	2	1	0	Reset
EXEN2	-	IEIIC	IELVI	IEKBI	IEADC	IESPI	IEPWM	00h

IEADC: A/D 轉換中斷致能位元

IEADC = 0 –致能 ADC 中斷.

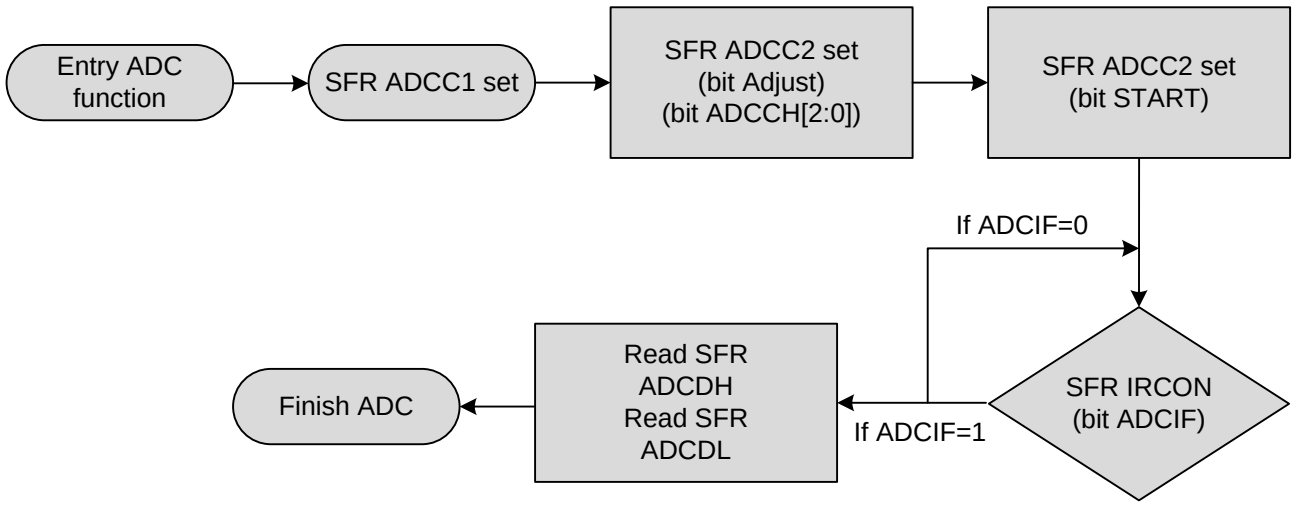
IEADC = 1 –致能 ADC 中斷.

Mnemonic: IRCON **Address: C0h**

7	6	5	4	3	2	1	0	Reset
EXF2	TF2	IICIF	LVIIF	KBIIF	ADCIF	SPIIF	PWMIF	00H

ADCIF: A/D 轉換中斷旗標位元，當有開啟 ADC 中斷時，轉換完成會設為 1，進中斷後硬體自動清為 0,若未開 ADC 中斷，則必須手動軟體清除為 0.

3.4.1 ADC一般應用流程圖

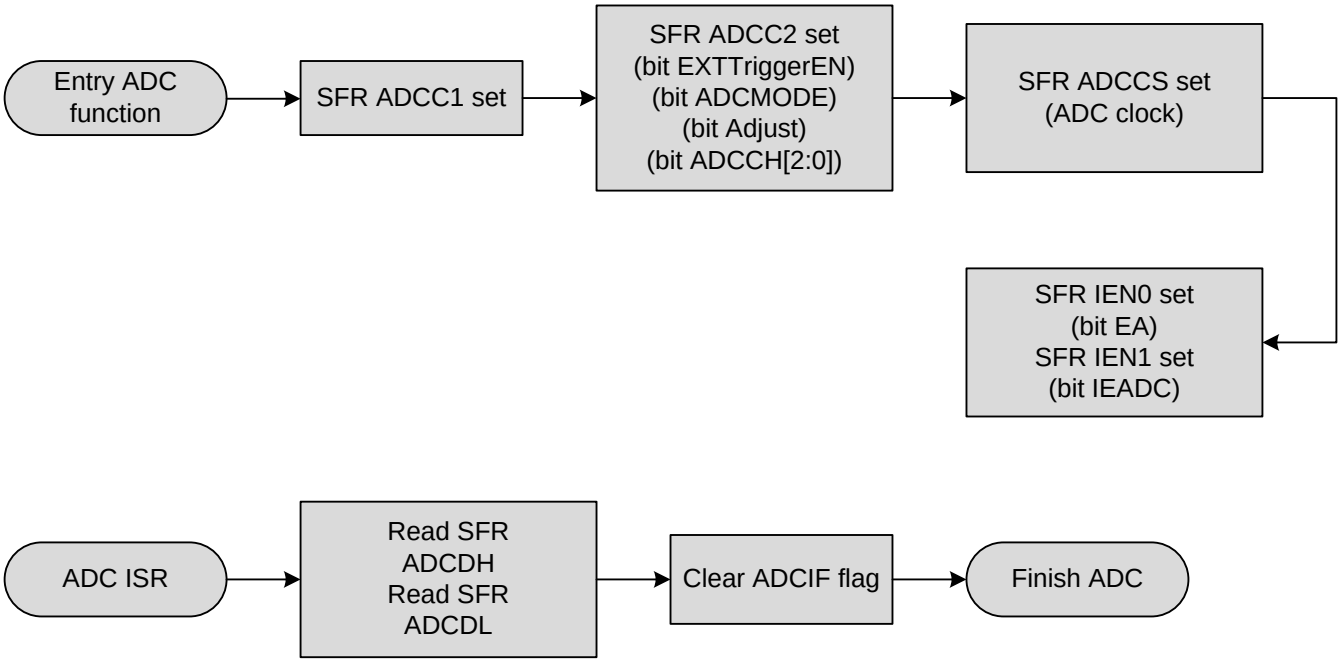


3.4.2 ADC一般程式範例

Describe:	Program:

main	//==========
	//
	// S Y N C M O S T E C H N O L O G Y
	//
	//==========
	#include "..\h\SM39A16M1.h"
	
	void main(void)
	{
	unsigned char temp_H,temp_L;
	
	PAGESEL = 0x01; //SFE Page0
	CKCON = 0x00; // Defult 2T(CKCON=0x10), Change to 1T. (CKCON SFR at //page 0)
	
	ADCC1 = 0x01; //ADC Chanel 0 enable
	ADCC2 = 0x00; //Continuous mode, ADC Chanel 0 is analog input, // Adiust=0
	ADCCS = 0x00; //ADC clock Fosc/2
	
	while(1)
	{
	ADCC2 |= 0x80; //sbit ADC START = 1, will auto clear after finish
	while(!IRCON && 0x04); //finish if ADCIF=1, converting if ADCIF=0
	temp_L = ADCDL; //ADC result, the Adjust=0, the ADCDL[1:0]=ADCD[1:0]
	temp_H = ADCDH; //the ADCDH[7:0]=ADCD[9:2]
	IRCON &= 0xFB; //Clear ADCIF flag for next ADC conversion
	}
	}

3.4.3 ADC外部觸發應用流程圖

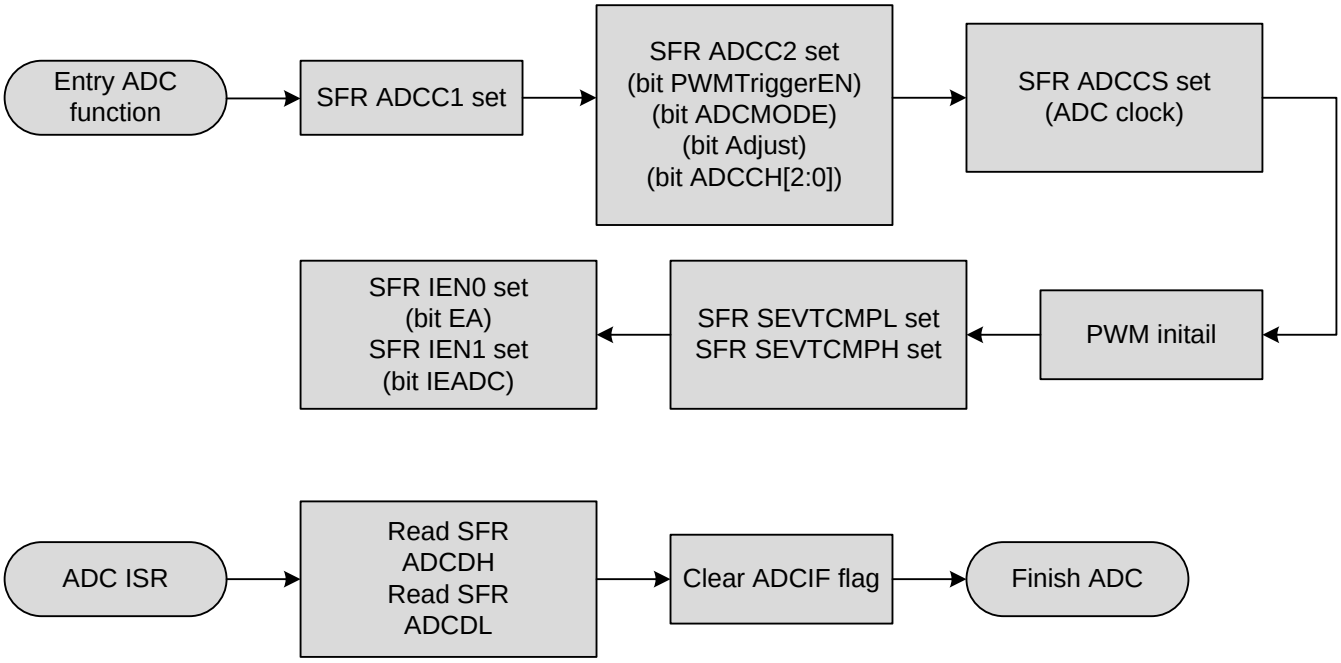


3.4.4 ADC外部觸發程式範例

Describe:	Program:

main	//==========
	//
	// SYNCMOS TECHNOLOGY
	//
	//==========
	#include "..\h\SM39A16M1.h"
	
	void main(void)
	{
	
	PAGESEL = 0x01; //SFE Page0
	CKCON = 0x00; // Defult 2T(CKCON=0x10), Change to 1T. (CKCON SFR at page 0)
	ADCC1 = 0x01; //ADC Chanel 0 enable
	ADCC2 = 0x18; //EXT TriggerADC Enable, single-shot mode, ADC Chane 0 is analog input, Adiust=0
	ADCCS = 0x00; //ADC clock Fosc/2
	
	EA = 1; //Enable all inierrupt
	IEADC = 1; //Enable ADC inierrupt
	while(1){}
	}
	
	void ADC_ISR(void) interrupt d_ADC_Vector // Address: 0x53
	{
	unsigned char temp_H,temp_L;
	
	temp_L = ADCDL; //ADC result, the Adjust=0, the ADCDL[1:0]=ADCD[1:0]
	temp_H = ADCDH; //the ADCDH[7:0]=ADCD[9:2]
	ADCIF =0; //Clear ADCIF flag for next ADC conversion
	}

3.4.5 PWM觸發ADC應用流程圖



3.4.6 PWM觸發ADC程式範例

Describe:	Program:

main	//==========
	//
	// SYNCMOS TECHNOLOGY
	//
	//==========
	#include "..\h\SM39A16M1.h"
	
	void main(void)
	{
	PAGESEL = 0x01; //SFE Page0
	CKCON = 0x00; // Defult 2T(CKCON=0x10), Change to 1T. (CKCON SFR at // page 0)
	ADCC1 = 0x01; //ADC Chanel 0 enable
	ADCC2 = 0x28; //PWM TriggerADC Enable, single-shot mode, ADC Chanel // 0 is analog input, Adiust=0
	ADCCS = 0x00; //ADC clock Fosc/2
	PAGESEL = 0x03; //SFE Page1
	SEVTCMPL = 0x00; //Special Event Compare Low byte
	SEVTCMPH = 0x00; //Special Event Compare High byte
	PAGESEL = 0x01; //SFE Page0
	PWM_Init();
	EA = 1; //Enable all inierrupt
	IEADC = 1; //Enable ADC inierrupt
	while(1){}
	}
	
	void ADC_ISR(void) interrupt d_ADC_Vector // Address: 0x53
	{
	unsigned char temp_H,temp_L;
	
	temp_L = ADCDL; //ADC result, the Adjust=0, the ADCDL[1:0]=ADCD[1:0]

```
        temp_H = ADCDH;          //the ADCDH[7:0]=ADCD[9:2]
        ADCIF =0;                //Clear ADCIF flag for next ADC conversion
}
```

3.5 SM39R08A2、SM39R16A2

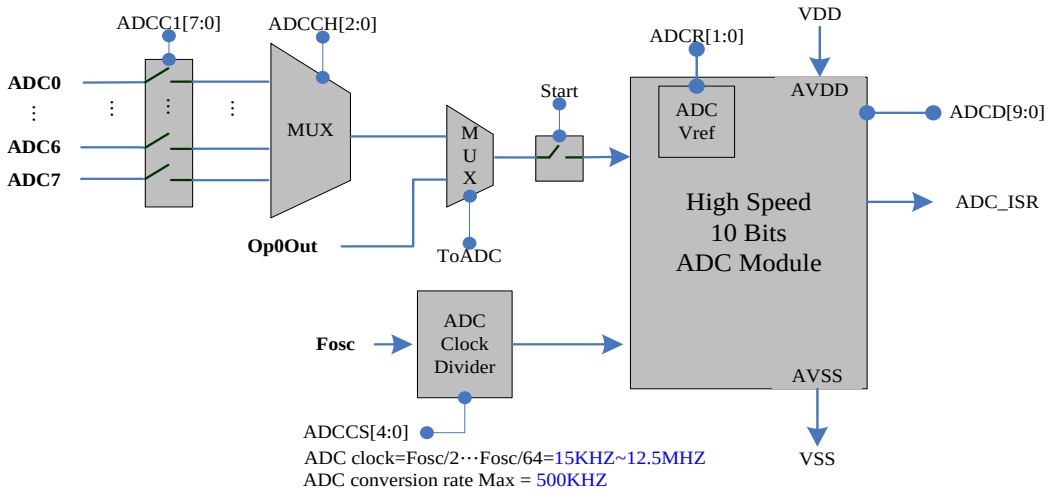


圖 3-4: ADC 模組工作方塊圖

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
ADC											
ADCC1	ADC Control register 1	ABh	ADC7EN	ADC6EN	ADC5EN	ADC4EN	ADC3EN	ADC2EN	ADC1EN	ADC0EN	00H
ADCC2	ADC Control register 2	ACh	Start	ADJUST	ADCR[1:0]		-	ADCCH[2:0]			00H
ADCDH	ADC data high byte	ADh	ADCDH [7:0]								00H
ADCDL	ADC data low byte	AEh	ADCDL [7:0]								00H
ADCCS	ADC clock select	AFh	-	-	-	ADCCS[4:0]					00H
IEN1	Interrupt Enable 1 register	B8h	EXEN2	-	IEIIC	IELVI	IEKBI	IEADC	IESPI	IEPWM	00h
IRCON	Interrupt request register	C0H	EXF2	TF2	IICIF	LVIIF	KBIIF	ADCIF	SPIIF	PWMIF	00H

Mnemonic: ADCC1 **Address: ABh**

7	6	5	4	3	2	1	0	Reset
ADC7EN	ADC6EN	ADC5EN	ADC4EN	ADC3EN	ADC2EN	ADC1EN	ADC0EN	00H

ADC7EN: 致能 ADC 通道 7.

ADC7EN = 1 –致能 ADC 通道 7

ADC6EN: 致能 ADC 通道 6.

ADC6EN = 1 –致能 ADC 通道 6

ADC5EN: 致能 ADC 通道 5.

ADC5EN = 1 –致能 ADC 通道 5

ADC4EN: 致能 ADC 通道 4.

ADC4EN = 1 –致能 ADC 通道 4

ADC3EN: 致能 ADC 通道 3.

ADC3EN = 1 –致能 ADC 通道 3

ADC2EN: 致能 ADC 通道 2.

ADC2EN = 1 –致能 ADC 通道 2

ADC1EN: 致能 ADC 通道 1.

ADC1EN = 1 –致能 ADC 通道 1

ADC0EN: 致能 ADC 通道 0.

ADC0EN = 1 –致能 ADC 通道 0

Mnemonic: ADCC2 **Address: ACh**

7	6	5	4	3	2	1	0	Reset
Start	ADJUST	ADCR[1:0]		-	ADCCH[2:0]			00H

Start: 當該位元被置位時，ADC 將啟動轉換.

ADJUST: ADC 數位輸出格式調整.

ADJUST = 0: (初始值)

ADC 數位元輸出高位元組 ADCD [9:2] = ADCDH [7:0].

ADC 數位輸出低位元組 ADCD [1:0] = ADCDL [1:0].

ADJUST = 1:

ADC 數位元輸出高位元組 ADCD [9:8] = ADCDH [1:0].

ADC 數位輸出低位元組 ADCD [7:0] = ADCDL [7:0].

ADCR[1:0]: ADC 轉換電壓區間選擇. (ADCR 為 write only，若 ADCR 設定 01 或 10 時，不可使用 ANL 或 ORL 去寫入 ADCC2)

ADCR [1:0]	**電壓區間**
00	0 ~ Vdd
01	$0 \sim \frac{4}{5} \times Vdd$
10	$0 \sim \frac{3}{5} \times Vdd$
11	保留

ADCCH[2:0]: ADC 通道選擇.

ADCCH [2:0]	**通道**
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

ADJUST = 0:

Mnemonic: ADCDH **Address: ADh**

7	6	5	4	3	2	1	0	Reset
ADCD[9]	ADCD[8]	ADCD[7]	ADCD[6]	ADCD[5]	ADCD[4]	ADCD[3]	ADCD[2]	00H

Mnemonic: ADCDL **Address: AEh**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	ADCD[1]	ADCD[0]	00H

ADJUST = 1:

Mnemonic: ADCDH **Address: ADh**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	ADCD[9]	ADCD[8]	00H

Mnemonic: ADCDL **Address: AEh**

7	6	5	4	3	2	1	0	Reset
ADCD[7]	ADCD[6]	ADCD[5]	ADCD[4]	ADCD[3]	ADCD[2]	ADCD[1]	ADCD[0]	00H

ADCD[9:0]: ADC 數字暫存器.

Mnemonic: ADCCS **Address: AFh**

7	6	5	4	3	2	1	0	Reset
-	-	-	ADCCS[4]	ADCCS[3]	ADCCS[2]	ADCCS[1]	ADCCS[0]	00H

ADCCS[4:0]: ADC Clock 選擇.

* ADC Clock 最大為 12.5MHz.
* ADC 轉換率最高為 500KHz

$$ADC\_Clock = \frac{\text{Fosc}}{2 \times (ADCCS + 1)}$$

$$ADC\_Conversion\_Rate = \frac{\text{ADC\_Clock}}{23}$$

Mnemonic: IEN1 **Address: B8h**

7	6	5	4	3	2	1	0	Reset
EXEN2	-	IEIIC	IELVI	IEKBI	IEADC	IESPI	IEPWM	00h

IEADC: A/D 轉換中斷致能位元

IEADC = 0 –致能 ADC 中斷.

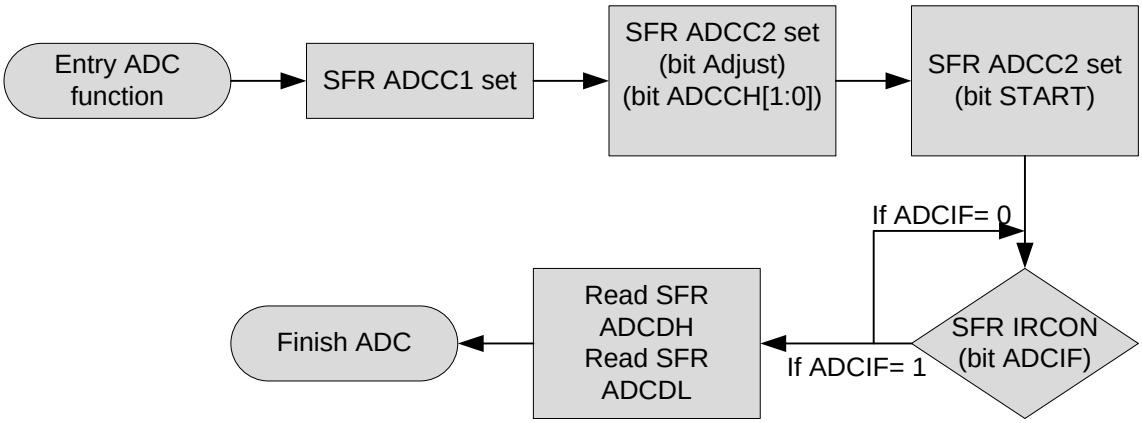
IEADC = 1 –致能 ADC 中斷.

Mnemonic: IRCON **Address: C0h**

7	6	5	4	3	2	1	0	Reset
EXF2	TF2	IICIF	LVIIF	KBIIF	ADCIF	SPIIF	PWMIF	00H

ADCIF: A/D 轉換中斷旗標位元，當有開啟 ADC 中斷時，轉換完成會設為 1，進中斷後硬體自動清為 0,未開 ADC 中斷，則必須手動軟體清除為 0.

3.5.1 ADC應用流程圖



3.5.2 ADC 程式範例

Describe:	Program:

main	//==========
	//
	// S Y N C M O S T E C H N O L O G Y
	//
	//==========
	#include "SM39R16A2.h"
	
	void main(void)
	{
	unsigned char temp_H,temp_L;
	
	ADCC1 = 0x01; //ADC Chanel 0 enable
	ADCC2 = 0x00; //ADC Chanel 0 is analog input, Adiust=0
	
	while(1)
	{
	ADCC2 =0x80; //sbit ADC START = 1, will auto clear after finish
	while(ADCIF); //finish if ADCIF=1, converting if ADCIF=0
	
	temp_L = ADCDL; //ADC result, the Adjust=0, the ADCDL[1:0]=ADCD[1:0]
	temp_H = ADCDH; //the ADCDH[7:0]=ADCD[9:2]
	ADCIF = 0; //Clear ADCIF flag for next ADC conversion
	}
	}

3.6 SM39R08A3、SM39R16A3

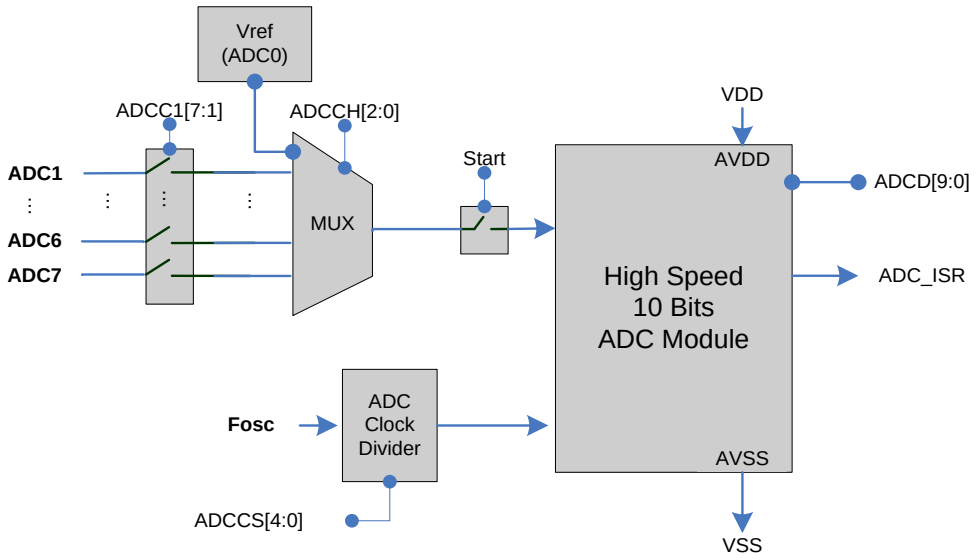


圖 3-5: ADC 模組工作方塊圖

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
ADC											
ADCC1	ADC Control register 1	ABh	ADC7 EN	ADC6 EN	ADC5 EN	ADC4 EN	ADC3 EN	ADC2 EN	ADC1 EN	ADC0E N	00H
ADCC2	ADC Control register 2	ACh	Start	ADJU ST	-		-	ADCCH[2:0]			00H
ADCDH	ADC data high byte	ADh	ADCDH [7:0]								00H
ADCDL	ADC data low byte	AEh	ADCDL [7:0]								00H
ADCCS	ADC clock select	AFh	-	-	-	ADCCS[4:0]					00H
IEN1	Interrupt Enable 1 register	B8h	EXEN 2	-	IEIIC	IELVI	IEKBI	IEADC	IESPI	IEP WM	00H
IRCON	Interrupt request register	C0H	EXF2	TF2	IICIF	LVIIF	KBIIF	ADCIF	SPIIF	PWM IF	00H

Mnemonic: ADCC1 **Address: ABh**

7	6	5	4	3	2	1	0	Reset
ADC7EN	ADC6EN	ADC5EN	ADC4EN	ADC3EN	ADC2EN	ADC1EN	ADC0EN	00H

ADC7EN: 致能 ADC 通道 7.

ADC7EN = 1 –致能 ADC 通道 7

ADC6EN: 致能 ADC 通道 6.

ADC6EN = 1 –致能 ADC 通道 6

ADC5EN: 致能 ADC 通道 5.

ADC5EN = 1 –致能 ADC 通道 5

ADC4EN: 致能 ADC 通道 4.

ADC4EN = 1 –致能 ADC 通道 4

ADC3EN: 致能 ADC 通道 3.

ADC3EN = 1 –致能 ADC 通道 3

ADC2EN: 致能 ADC 通道 2.

ADC2EN = 1 –致能 ADC 通道 2

ADC1EN: 致能 ADC 通道 1.

ADC1EN = 1 –致能 ADC 通道 1

Mnemonic: ADCC2 **Address: ACh**

7	6	5	4	3	2	1	0	Reset
Start	ADJUST	-		-	ADCCH[2:0]			00H

Start: 當該位元被置位時，ADC 將啟動轉換.

ADJUST: ADC 數位輸出格式調整.

ADJUST = 0: (初始值)

ADC 數位元輸出高位元組 ADCD [9:2] = ADCDH [7:0].

ADC 數位輸出低位元組 ADCD [1:0] = ADCDL [1:0].

ADJUST = 1:

ADC 數位元輸出高位元組 ADCD [9:8] = ADCDH [1:0].

ADC 數位輸出低位元組 ADCD [7:0] = ADCDL [7:0].

ADCCH[2:0]: ADC 通道選擇.

ADCCH [2:0]	通道
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

切換至 ADC0 通道後, 即可讀取來自內部 Vref 1.2V±10%,無外部輸入 Pin 腳.

ADJUST = 0:

Mnemonic: ADCDH **Address: ADh**

7	6	5	4	3	2	1	0	Reset
ADCD[9]	ADCD[8]	ADCD[7]	ADCD[6]	ADCD[5]	ADCD[4]	ADCD[3]	ADCD[2]	00H

Mnemonic: ADCDL **Address: AEh**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	ADCD[1]	ADCD[0]	00H

ADJUST = 1:

Mnemonic: ADCDH **Address: ADh**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	ADCD[9]	ADCD[8]	00H

Mnemonic: ADCDL **Address: AEh**

7	6	5	4	3	2	1	0	Reset
ADCD[7]	ADCD[6]	ADCD[5]	ADCD[4]	ADCD[3]	ADCD[2]	ADCD[1]	ADCD[0]	00H

ADCD[9:0]: ADC 數字暫存器.

Mnemonic: ADCCS **Address: AFh**

7	6	5	4	3	2	1	0	Reset
-	-	-	ADCCS[4]	ADCCS[3]	ADCCS[2]	ADCCS[1]	ADCCS[0]	00H

ADCCS[4:0]: ADC Clock 選擇.

* ADC Clock 最大為 12.5MHz.
* ADC 轉換率最高為 961KHz

$$ADC\_Clock = \frac{\text{Fosc}}{2 \times (ADCCS + 1)}$$

$$ADC\_Conversion\_Rate = \frac{\text{ADC\_Clock}}{13}$$

Mnemonic: IEN1 **Address: B8h**

7	6	5	4	3	2	1	0	Reset
EXEN2	-	IEIIC	IELVI	IEKBI	IEADC	IESPI	IEPWM	00h

IEADC: A/D 轉換中斷致能位元

IEADC = 0 –致能 ADC 中斷.

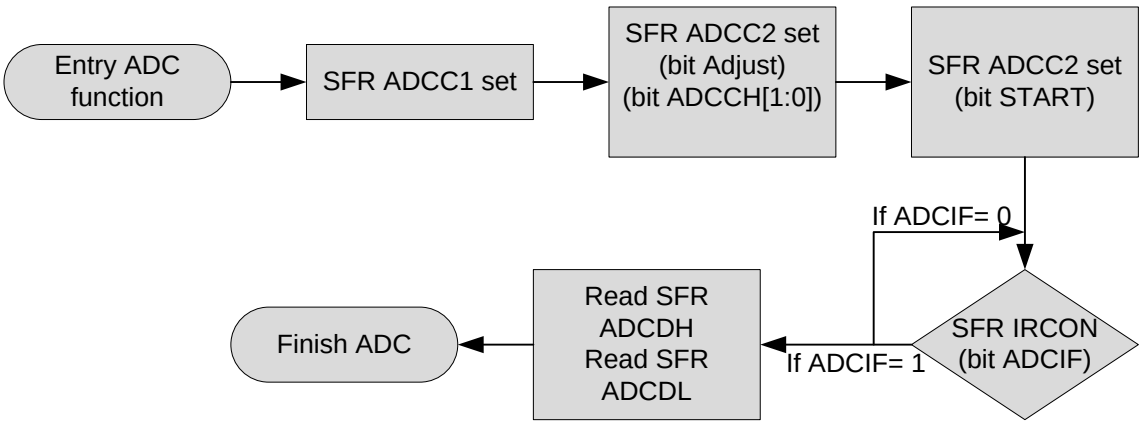
IEADC = 1 –致能 ADC 中斷.

Mnemonic: IRCON **Address: C0h**

7	6	5	4	3	2	1	0	Reset
EXF2	TF2	IICIF	LVIIF	KBIIF	ADCIF	SPIIF	PWMIF	00H

ADCIF: A/D 轉換中斷旗標位元，當有開啟 ADC 中斷時，轉換完成會設為 1，進中斷後硬體自動清為 0, 若未開 ADC 中斷，則必須手動軟體清除為 0.

3.6.1 ADC應用流程圖



3.6.2 ADC程式範例

說明: 利用內部 Vref 去換算外部電池電壓方式，可省去外部參考電源(TL-431)

Describe:	Program:

main	//==========
	//
	// S Y N C M O S T E C H N O L O G Y
	//
	//==========
	#include "SM39R16A3.h"
	
	#define Vref 1.22 //only for A3
	#define d_ADC_CH0_IN 0
	#define d_ADC_CH1_IN 1
	#define d_ADC_CLK_DIV6 0x02
	
	unsigned int uiVoltage = 0,uiVref = 0,
	
	void ADCInit(unsigned char n_ADC_CLK)
	{
	ADCCS = n_ADC_CLK; //Select ADC clock
	}
	
	void ADCstart(void)
	{
	ADCIF = 0;
	ADCC2 = ADCC2|0xC0; //ADC start conversion
	}
	void ADCChannel(unsigned char n_ADC_CH)
	{
	ADCC2 = n_ADC_CH; //Set conversion channel
	}
	
	void ADC_stop(void)
	{
	ADCC2 &= 0x7F; //ADC stop conversion
	}
	unsigned int ADC_Channel(unsigned char c_ADC_channel)
	{
	unsigned int uiADC10;
	
	switch(c_ADC_channel&0xFF)

```
        {
            case 0:
                ADCC1 = 0x01; //Set ADC channel
                break;

            case 1:
                ADCC1 = 0x02; //Set ADC channel
                break;

            case 2:
                ADCC1 = 0x04; //Set ADC channel
                break;

            case 3:
                ADCC1 = 0x08; //Set ADC channel
                break;

            case 4:
                ADCC1 = 0x10; //Set ADC channel
                break;

            case 5:
                ADCC1 = 0x20; //Set ADC channel
                break;

            case 6:
                ADCC1 = 0x40; //Set ADC channel
                break;

            case 7:
                ADCC1 = 0x80; //Set ADC channel
                break;
        }
        ADCChannel(c_ADC_channel);
        ADCstart();
        while(!ADCIF);
        uiADC10 = (ADCDH * 256)+ ADCDL;
        ADC_stop();

        ADCC1 = 0x00; //Set ADC channel

        return uiADC10
}

unsigned int Check_Vref_Voltage(unsigned char ADC_ch)   //VREF
{
        unsigned int uiBAT;
        float ftemp;

        ftemp=(ADC_Channel(ADC_ch));
        uiBAT=(unsigned int)ftemp;
        return uiBAT;
}

unsigned int Check_Battery_Voltage(unsigned char ADC_ch)      //BAT_VOL_DET
{
        unsigned int uiBAT;
        float ftemp;

        ftemp=((Vref*(ADC_Channel(ADC_ch)))/(uiVref))*1000;
        uiBAT=(unsigned int)ftemp;
```

```
        return uiBAT;
}

void main(void)
{

        ADCInit(d_ADC_CLK_DIV6);
        while(1)
        {
                uiVref=Check_Vref_Voltage(d_ADC_CH0_IN);    //only for A3 internal 1.2V
                uiVoltage=Check_Battery_Voltage(d_ADC_CH1_IN);
        }
}
```

3.7 SM39R08A5

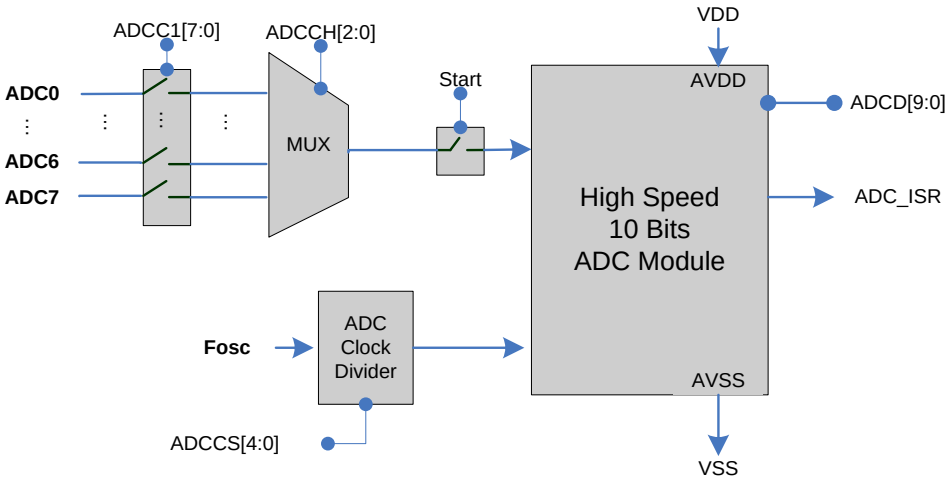


圖 3-6: ADC 模組工作方塊圖

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
ADC											
ADCC1	ADC Control register 1	ABh	ADC7 EN	ADC6 EN	ADC5E N	ADC4 EN	ADC3 EN	ADC2 EN	ADC1 EN	ADC0 EN	00H
ADCC2	ADC Control register 2	ACh	Start	ADJUST	-	-	-	ADCCH[2:0]			00H
ADCDH	ADC data high byte	ADh	ADCDH [7:0]								00H
ADCDL	ADC data low byte	AEh	ADCDL [7:0]								00H
ADCCS	ADC clock select	AFh	-	-	-	ADCCS[4:0]					00H
IEN1	Interrupt Enable 1 register	B8h	EXEN2	-	IEIIC	IELVI	IEKBI	IEADC	IESPI	IEPWM	00h
IRCON	Interrupt request register	C0H	EXF2	TF2	IICIF	LVIIF	KBIIF	ADCIF	SPIIF	PWMIF	00H

Mnemonic: ADCC1 **Address: ABh**

7	6	5	4	3	2	1	0	Reset
ADC7EN	ADC6EN	ADC5EN	ADC4EN	ADC3EN	ADC2EN	ADC1EN	ADC0EN	00H

ADC7EN: 致能 ADC 通道 7.

ADC7EN = 1 –致能 ADC 通道 7

ADC6EN: 致能 ADC 通道 6.

ADC6EN = 1 –致能 ADC 通道 6

ADC5EN: 致能 ADC 通道 5.

ADC5EN = 1 –致能 ADC 通道 5

ADC4EN: 致能 ADC 通道 4.

ADC4EN = 1 –致能 ADC 通道 4

ADC3EN: 致能 ADC 通道 3.

ADC3EN = 1 –致能 ADC 通道 3

ADC2EN: 致能 ADC 通道 2.

ADC2EN = 1 –致能 ADC 通道 2

ADC1EN: 致能 ADC 通道 1.

ADC1EN = 1 –致能 ADC 通道 1

ADC0EN: 致能 ADC 通道 0.

ADC0EN = 1 –致能 ADC 通道 0

Mnemonic: ADCC2 **Address: ACh**

7	6	5	4	3	2	1	0	Reset
Start	ADJUST	-	-	-	ADCCH[2:0]			00H

Start: 當該位元被置位時，ADC 將啟動轉換.

ADJUST: ADC 數位輸出格式調整.

ADJUST = 0: (初始值)

ADC 數位元輸出高位元組 ADCD [9:2] = ADCDH [7:0].

ADC 數位輸出低位元組 ADCD [1:0] = ADCDL [1:0].

ADJUST = 1:

ADC 數位元輸出高位元組 ADCD [9:8] = ADCDH [1:0].

ADC 數位輸出低位元組 ADCD [7:0] = ADCDL [7:0].

ADCCH[2:0]: ADC 通道選擇.

ADCCH [2:0]	Channel
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

ADJUST = 0:

Mnemonic: ADCDH **Address: ADh**

7	6	5	4	3	2	1	0	Reset
ADCD[9]	ADCD[8]	ADCD[7]	ADCD[6]	ADCD[5]	ADCD[4]	ADCD[3]	ADCD[2]	00H

Mnemonic: ADCDL **Address: AEh**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	ADCD[1]	ADCD[0]	00H

ADJUST = 1:

Mnemonic: ADCDH **Address: ADh**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	ADCD[9]	ADCD[8]	00H

Mnemonic: ADCDL **Address: AEh**

7	6	5	4	3	2	1	0	Reset
ADCD[7]	ADCD[6]	ADCD[5]	ADCD[4]	ADCD[3]	ADCD[2]	ADCD[1]	ADCD[0]	00H

ADCD[9:0]: ADC 數字暫存器.

Mnemonic: ADCCS **Address: AFh**

7	6	5	4	3	2	1	0	Reset
-	-	-	ADCCS[4]	ADCCS[3]	ADCCS[2]	ADCCS[1]	ADCCS[0]	00H

ADCCS[4:0]: ADC Clock 選擇.

* ADC Clock 最大為 11.0592MHz.
* ADC 轉換率最高為 851KHz

$$ADC\_Clock = \frac{Fosc}{2 \times (ADCCS + 1)}$$

$$ADC\_Conversion\_Rate = \frac{ADC\_Clock}{13}$$

Mnemonic: IEN1 **Address: B8h**

7	6	5	4	3	2	1	0	Reset
EXEN2	-	IEIIC	IELVI	IEKBI	IEADC	IESPI	IEPWM	00h

IEADC: A/D 轉換中斷致能位元

IEADC = 0 –致能 ADC 中斷.

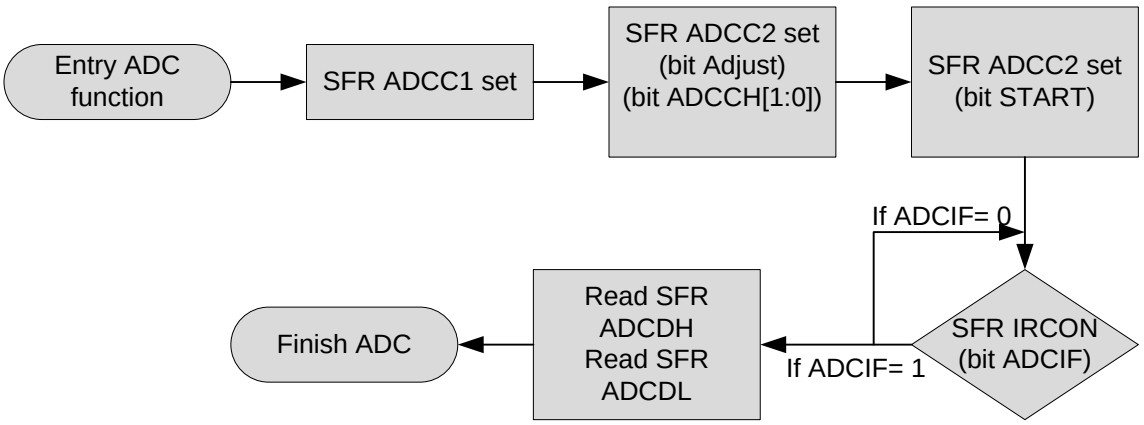
IEADC = 1 –致能 ADC 中斷.

Mnemonic: IRCON **Address: C0h**

7	6	5	4	3	2	1	0	Reset
EXF2	TF2	IICIF	LVIIF	KBIIF	ADCIF	SPIIF	PWMIF	00H

ADCIF: A/D 轉換中斷旗標位元，當有開啟 ADC 中斷時，轉換完成會設為 1，進中斷後硬體自動清為 0, 若未開 ADC 中斷，則必須手動軟體清除為 0.

3.7.1 ADC應用流程圖

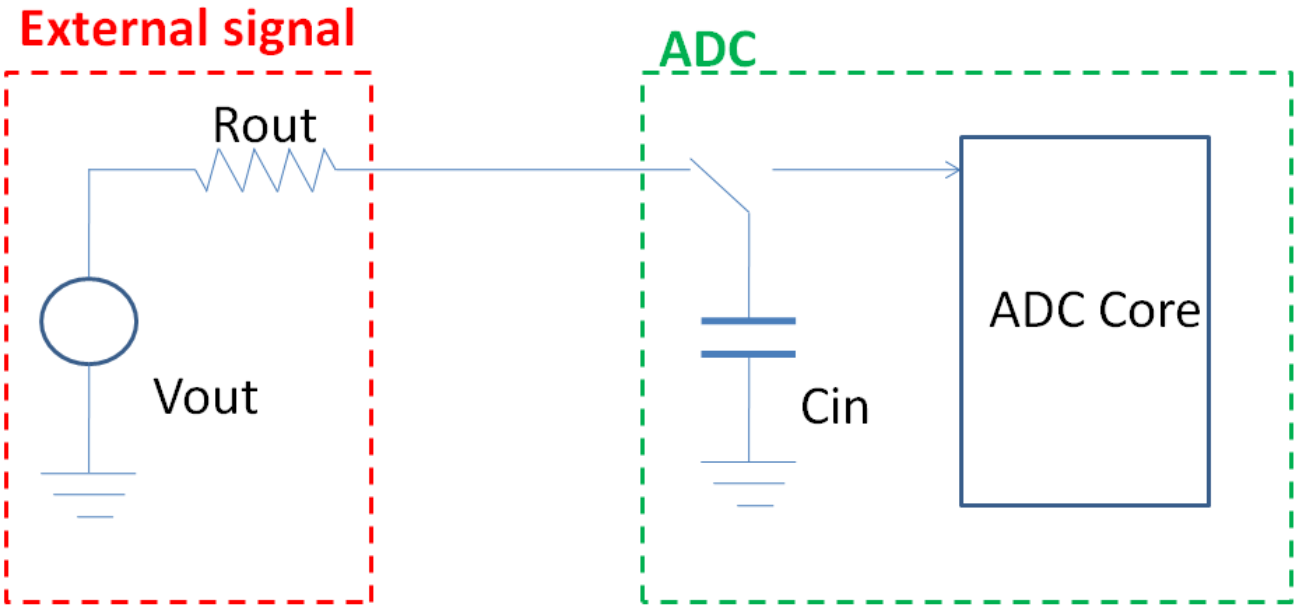


3.7.2 ADC程式範例

Describe:	Program:

main	//===
	//
	// S Y N C M O S T E C H N O L O G Y
	//
	//===
	#include "..\h\SM59R08A5.h"
	
	void main(void)
	{
	unsigned char temp_H,temp_L;
	
	ADCC1 = 0x01; //ADC Chanel 0 enable
	ADCC2 = 0x00; //ADC Chanel 0 is analog input, Adiust=0
	
	while(1)
	{
	ADCC2 |=0x80; //sbit ADC START = 1, will auto clear after finish
	while(!IRCON && 0x04); //finish if ADCIF=1, converting if ADCIF=0
	
	temp_L = ADCDL;/ /ADC result, the Adjust=0, the ADCDL[1:0]=ADCD[1:0]
	temp_H = ADCDH; //the ADCDH[7:0]=ADCD[9:2]
	IRCON &= 0xFB; //Clear ADCIF flag for next ADC conversion
	}
	}

4. 最佳ADC取樣頻率選擇說明



適用型號：SM39R16A3、SM39R08A3、SM39R08A5、SM39R16A6、SM39A16M1、SM59A16U1。

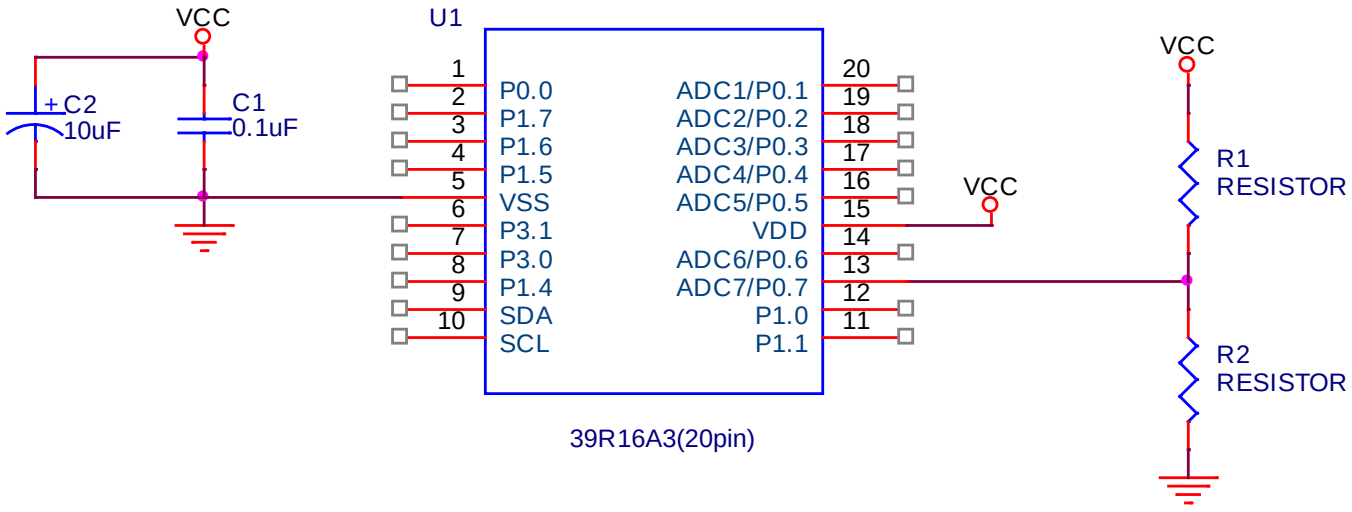
Rout 阻值的選擇必須符合下列公式：

$$Rout < 3.33 * Ta * 10^{10}$$

Ta: $\frac{1}{\text{ADC clock}}$

範例 1：

Fosc = 22.1184MHz, ADCCS[4:0] = Fosc/2.



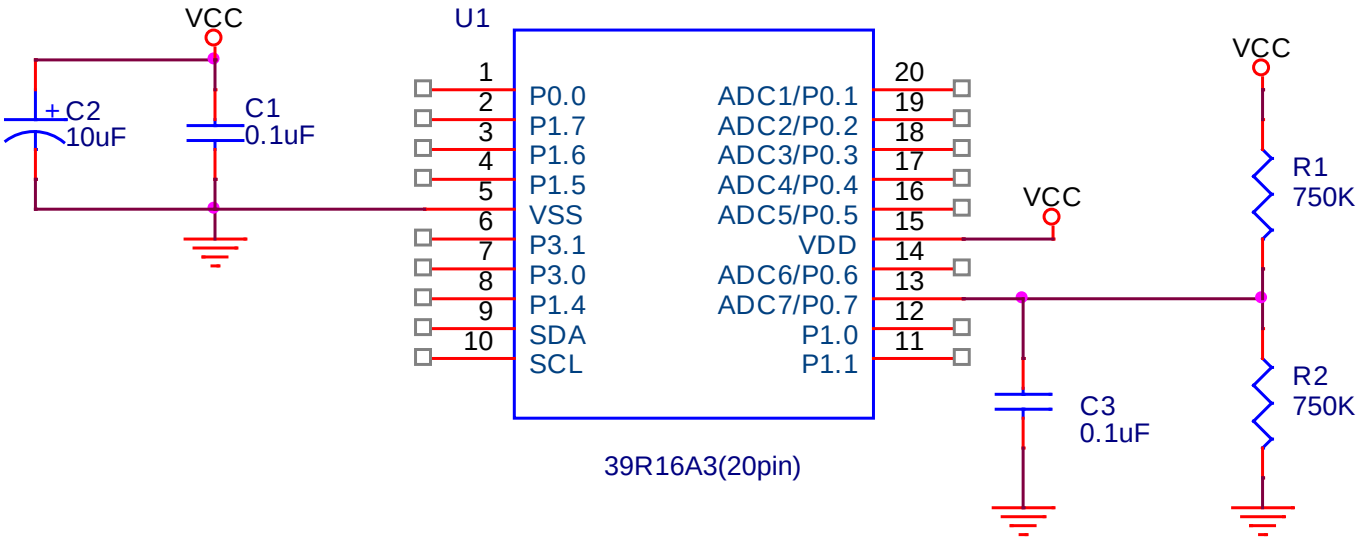
Rout < 3.33*(1/11.0592MHz)* 10^{10}

Rout < 3011Ω

因 Rout = R1//R2 (並聯等校阻抗)，
所以若 R1 與 R2 使用相同阻值，則 R1 與 R2 阻值以小於 1505Ω 為宜。

範例 2：

當客戶使用於電池應用時，基於省電考量，ADC 阻值會使用較大的阻值來達到省電目的，導致無法符合 Rout 公式的要求。這時可在 ADC 通道多加一個 0.1uF，來達成 ADC 的精準度需求。



5. 注意事項

每一顆零件號碼因頻率、通道或設計上不同，使用上可能會有些差異，請使用者務必參照上述的不同群組的歸類方式來設定所需的內容值。