第七章

即時時鐘

7.1 即時時鐘（Real-Time Clock，RTC）

即時時鐘（RTC）在一般個人電腦算是系統提供的時鐘電路，為減少個人電腦 CPU 的工作負擔因此將此 RTC 的電路功能移出系統核心由一個專用的 RTC 積體電路負責維持時鐘計時功能，RTC 常用時脈來源是由 32.768KHz 的振盪電路提供，而此時脈來源與電一般的石英錶是一樣的，大約 2^{15} 個時脈計時一秒。因此在 RTC 單元常見的工作時脈頻率就是 32.768KHz，當然也有其他的選擇，比如利用電力線信號頻率的。市面上供應許多 RTC 晶片，Motorola MC146818 就是一個提供萬年曆功能的 RTC 晶片，早期個人電腦採用它來當即時時鐘，後來 Dallas 公司提供許多 RTC 晶片其中以 DS 系列最多。

7.2 WT59F064 RTC

7.2.1 WT59F064 RTC 特性與使用說明

WT59F064 RTC 占用位址空間為 0x00201000 ~ 0x002013FF，在使用前必須先設定系統暫存器（0x200018）的 RTC_CS、RTC_EN 等位元啟動 RTC 晶片的使用，同時設定系統暫存器（0x200020）LSEON 位元選用外部 32.768KHz 的時脈。

當 32.768KHz 時脈設定完畢後緊著可以透過清除 RTC_PDOSC、RTC_STOP 等二位元達到啟動計時的功能，至於計時初值年、月、日、星期、時、分、秒等設定可隨時寫入到 0x00201000 ~ 0x00201018 等位址上。但是需注意到寫入或讀出的資料一律是 BCD 的編碼格式，例如要設定日期 23 號，就必須將 0x23 送入 RTC_DAY（0x20100C）暫存器，而不是送入 0x17，但是讀出的日期 0x23 時，就是代表 23 日。

表 7.2.1-1 WT59F064 RTC 暫存器列表（位址：0x00201000 ~ 0x002013FF）

索引位址	預設值	讀／寫	位元	名稱	功能設定
+0x00	0	R/W	6:0	RTC_SEC	Second 秒（以 BCD 碼格式儲存），範圍 0~59 SEC [6:4] 秒的十位數 SEC[3:0] 秒的個位數
+0x04	0	R/W	6:0	RTC_MIN	Minute 分（以 BCD 碼格式儲存），範圍 0~59 MIN[6:4] 分的十位數 MIN[3:0] 分的個位數
+0x08	0	R/W	5:0	RTC_HOUR	Hour 時（以 BCD 碼格式儲存），範圍 0~23 HOUR[5:4] 時的十位數 HOUR[3:0] 時的個位數
+0x0C	01	R/W	5:0	RTC_DAY	日(以 BCD 碼格式儲存)，範圍 1~31 DAY[5:4] 日的十位數 DAY[3:0] 日的十位數 s
+0x10	0	R/W	2:0	RTC_WEEK	Day of week（星期） 000：週日，001：週一，010：週二，011：週三，100：週四，101：週五，110：週六
+0x14	01	R/W	3:0	RTC_MONTH	Month 月份 0001：一月，0010：二月，0011：三月，0100：四月，0101：五月，0110：六月，0111：七月，1000：八月，1001：九月，1010：十月，1011：十一月，1100：十二月。
+0x18	0x200	R/W	7:0	RTC_YEAR	Year 年（以 BCD 碼格式儲存），範圍 0~99 YEAR[7:4] 年份的十位數 YEAR [3:0] 年份的個位數
+0x20	0x200	R/W	7:0	RTC_BAKUP1	備份暫存器 1
+0x24	0x200	R/W	7:0	RTC_BAKUP2	備份暫存器 2
+0x28	0x200	R/W	7:0	RTC_BAKUP3	備份暫存器 3
+0x2C	0x3FF	R/W	7:0	RTC_BAKUP4	備份暫存器 4
+0x30	0x3FF	R/W	3:0	AMP	Amplifier stages 調整電流 AMP[3:2]：增強第二個放大器電流 AMP[1:0]：增強第一個放大器電流

索引位址	預設值	讀／寫	位元	名稱	功能設定
					「11」: four units ,「10」: three units 「01」: two units ,「00」: one unit
+0x34	0x3FF	R/W	7:0	RTC_CA	Calibration bits，校準位元 CA[7]=1，增加時脈，CA[7]=0，減少時脈 CA[6:0] =在 128 分中增加或減少時脈的個數（ add/skip 128 crystal clocks in one second of minute）
+0x38	0x1	R/W	7	RTC_ PDOSC	0：啟動 32.768KHz 振盪器 1：停止 32.768KHz 振盪器以節能
	0	R/W	6	RTC_STOP	0：啟動 RTC 計時功能 1：停止 RTC 計時
	0	R/W	5	RTC_FAST	1：RTC 快速測試模式 0：一般模式
	0	R/W	4	RTC_TEST	0：RTC 計數時脈由振盪子取得 1：RTC 測試模式，由 I/O 接腳輸入高速時脈加快計時
	0	R/W	3	RTC_PDOSCSU	0：Default 1：停止 32768Hz 振盪電路以節能
	0x1	R/W	2:0	RTC_FS	Clock output frequency，RTC 輸出頻率 000：No output，001：0.25Hz，010：1Hz 011：8Hz，100：64Hz，101：512Hz 110：1024Hz，111：32768Hz
+0x3C	0x6	R/W	7:4	RX	Bias resistor selection 偏壓電阻選擇 0000：50k，0001：300k，0010：350k 0011：400k，0100：450k，0101：500k 0110：550k（預設），0111：600k，1000：650k
	0x1	R/W	1	DRV2	Crystal bias current control 1：Default 0：Enlarge the crystal bias current
	0	R/W	0	DRV1	振盪電路驅動增益控制 1：強化增益 0：預設

7.2.3 WT59F064 RTC 實作

- **ADP-WT59F064-RTC 實作**

此實作的主要目的是利用 RTC 特性提供一數位時鐘，可以顯示年月日星期及時分秒，因此此一實作將使用 RTC 與 LCM 等兩個元件，LCM 只是用來顯示執行成果，LCM 的使用在第四章通用輸出入單元中已介紹不再重複。整個 ADP-WT59F064 專案的重點在於 main.c 中，尤其中可看出進入 main( ) 中先進行 LCM[LCMInit( )] 及 RTC[InitialRTC( )] 的初始化，在 InitialRTC( ) 中只是簡單地設定 RTC 計時的時脈來源及啟動 RTC 及設定其為可存取。設定指令如下：

```
//啟動 RTC 晶片致能與存取功能
System_Control_18 = ( System_Control_18 | 0x0060);
//啟動外部 32.768KHz2k 的時脈
System_Control_20 = ( System_Control_20 | 0x0005);
```

System_Control_20 與 System_Control_18（參考表 7.2.1-1）是 APB 上的兩個系統控制（System Control）暫存器，系統控制暫存器共有 15 個，占用的位址空間為 0x00200000 ~ 0x002003FF，其中索引位址 +18 的暫存器是控制 RTC 的暫存器，分別設定位元 RTC_CS=1、RTC_EN=1，並且啟動外部 32.768KHz 的震盪器以供計時用。此外設定「RTC_CTR=0x07」啟動 32.768KHz 震盪器及計時。初始化完成後緊接著設定年、月、日、星期、時、分、秒等資料，設定時需以 BCD 碼格式設定，實際上當執行完 InitialRTC( )，RTC 已開始計時，當以上日期、時間寫入 RTC_YEAR、RTC_MONTH、……、RTC_SEC 等暫存器不響計時，RTC 會依照新的時間、日期繼續計時，接著 main( ) 呼叫 ShowTime( ) 將 RTC 內 RTC_YEAR、RTC_MONTH、RTC_DAY 等顯示在 LCM 第一列，RTC_HOUR、RTC_MIN、RTC_SEC 等顯示在 LCM 第二列，因為從以上暫存器讀出的資料已是 BCD

碼，因此只須將每個數字的 ASCII 碼送入 LCM 資料暫存器即可，以秒為例說明程式碼：

```
sec = RTC_SEC;
while (LCMCheckBusyAdr() & 0x80);
//取得秒數的十位數加 0X30 轉換成 ASCII 送到 LCM
LCMDATAWR(((sec >> 4) & 0x000F) + To_ASCII);
while (LCMCheckBusyAdr() & 0x80);
//取得秒數的個位數加上 0X30 轉換成 ASCII 送到 LCM
LCMDATAWR((sec & 0x000F) + To_ASCII);
```

表 7.2.2-1 部分系統暫存器（位址：0x0020_0000 ~ 0x0020_03FF）

索引位址	預設值	讀／寫	位元	名稱	功能設定
+0x18	0	R	7	RTC_1S	1：Event of RTC 1s 0：No event of RTC 1s
	0	R/W	6	RTC_CS	1：Enable chip select of RTC WR/RD 0：Disable chip select of RTC WR/RD
	0	R/W	5	RTC_EN	1：Enable access RTC 0：Disable RTC
	0	W	0	CLR_RTC_1S	1：Clear event RTC 1s interrupt 0：No clear event RTC 1s interrupt
+0x20	0	R/W	4:3	LDO_OFF	LDO_OFF[1]: power down major LDO directly LDO_OFF[0]: power down major LDO during suspend mode
	0	R/W	2	LSEON	External low speed (32K) XTAL enable
	1	R/W	1	LSION	Internal low speed (128K) oscillator enable
	1	R/W	0	TS_PD	Temperature sensor power down 0：temperature sensor on 1：temperature sensor off

以下是匯入專案執行步驟：

Step 1. 將滑鼠移入專案瀏覽（Project Viewer）子視窗中點擊滑鼠右鍵後，選 import 進行匯入「ADP-WT59F064-RTC」專案。

Step 2. 打開 General 選擇「 Existing Projects into Workspace 」後，按 Next > 。

Step 3. 因為匯入的是專案壓縮檔因此點擊「 Select archive file 」後再打開點擊「 Browse 」選項選擇要匯入的專案。

Step 4. 點選「 ADP-WT59F064-RTC.zip 」這個專案壓縮檔後，按開啟舊檔 (O)。

Step 5. 開啟專案壓縮檔後可見到專案檔的完整名稱（ ADP-WT59F064-RTC ），在專案檔名稱左方勾選者表示要匯入此專案，再點擊下方「 Finish 」選項。

Step 6. 透過「 Build Project 」編譯專案檔裡的程式碼。先以滑鼠左鍵點選「 ADP-WT59F064-RTC 」使其出現反白，然後按滑鼠右鍵後再點選「 Build Project 」進行編譯連結的工作。

Step 7. 把 Step 6. 中建置完成的「 ADP-WT59F064-RTC.bin 」檔透過燒錄程式 ISP_WT59F064.exe 燒錄到晶片中。在專案瀏覽子視窗中對「 ADP-WT59F064- RTC 」按右鍵出現下拉式選單，再點擊「 Flash Burner 」以開啟燒錄視窗。

Step 8. 燒錄程式視窗中可以見到燒錄程式檔名，要燒錄到晶片中的程式碼存放路徑及檔案名稱，燒錄前請再次確認是否正確。透過點擊「 Auto 」一次完成整個燒錄過程，當出現「 Verify Successful 」，表示燒錄成功。

Step 9. 程式執行時可以到 LCM 第一列出「 年-月-日 」，第二列則是出現 24 小時制的「 時：分：秒 」，也是在 LCM 上呈現一個數位時鐘。圖 7.2.2-1 是 ADP-WT59F064 的執行結果。

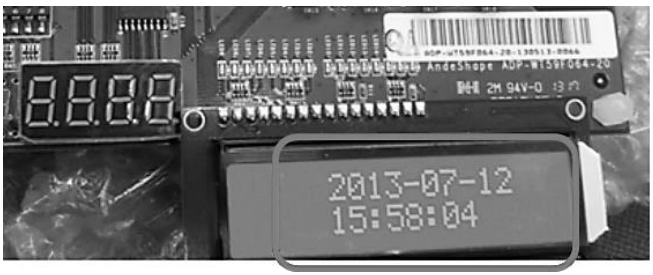

圖 7.2.2-1 ADP-WT59F064 專案執行結果

7.3 ADP-XC5-for-N801-S Real-Time Clock（RTC）

7.3.1 ADP-XC5-for-N801-S RTC 特性

ADP-XC5-for-N801-S 提供的 RTC 採智原科技 FTRTC010，工作時使用兩種時脈來源，一種是周邊匯流排系統時脈（PCLK），另一種是 EXTCLK（通常是 1Hz 的時脈或是其他），後面一種主要是提供給計時器使用。FTRTC010 提供日、時、分、秒各別的計數器，除可減少軟體的複雜度外，也可降低功率消耗。FTRTC010 也提供自動鬧鈴（auto alarm）功能，只要啟動秒的鬧鈴功能，每秒都會產生一次中斷請求信號，日、時、分也同樣具有相同的自動鬧鈴功能。

FTRTC010 其特性簡述如下：

- 當系統進入休眠狀態，可將 PCLK 切斷以節能。
- 提供各別日、時、分、秒的計數器。
- 提供各別日、時、分、秒自動鬧鈴功能。

7.3.2 ADP-XC5-for-N801-S RTC 特性使用設定

表 7.2.1-1 列出 FTRTC010 所有的暫存器，RtcSecond、RtcMinute、RtcHour、RtcDays 等分別是秒、分、時、日的計時計數器，是只能讀出的暫存器，要查詢目前的時間就是從這裡讀取，RtcSecond、RtcMinute、RtcHour 每個暫存器存值的範圍分別 0 ~ 59、0 ~ 59、0 ~ 23。RtcSecond 每秒加 1 只要超過 59 就歸 0，RtcMinute 每分加 1 只要超過 59 就歸 0、RtcHour 每個小時加 1 只要超過 23 就歸 0，RtcDays 則每天加 1。要設定以上暫存器的值時需先將值分別寫入 WRtcSecond、WRtcMinute、WRtcHour、WRtcDays 等暫存器中，等到 RtcCR[6] 被設為 1 時這些值會各別被寫入 RtcSecond、RtcMinute、RtcHour、RtcDays。

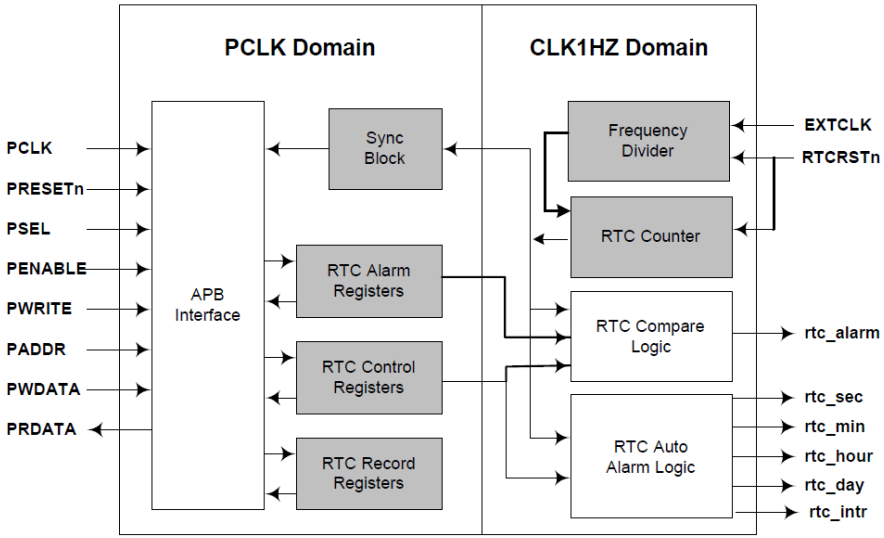

圖 7.3.2-1 FTRTC010 方塊圖（擷取自 ATFRTC010_DS_v1.2.pdf）

AlarmSecond、AlarmMinute、AlarmHour 等三個暫存器用來設定鬧鈴時間，等到跟 RtcSecond、RtcMinute、RtcHour 等三個暫存器完全相符時就透過 rtc_alarm 發出中斷請求，前提是 RtcCR[5]=1。要將 rtc_alarm 送出 RTC 外部時需先設定 RtcCR[0]=1。此外，如果要設定自動整點鬧鈴功能，可以設定 RtcCR[1] ~ RtcCR[4]，如此每秒、每分、每時、每日皆可發出一個中斷請求信號。中斷產生時，可藉由讀取中斷狀態暫存器（表 7.3.2-3）來判斷是哪種中斷發生，進而進行相關動作。

表 7.3.2-1 ADP-XC5-for-N801-S RTC 暫存器列表（基底位址：0x00201000）

索引位址	R/W	位元寬度	名稱	設定說明	預設值
+0x00	R	6	RtcSecond	RTC second register，秒計數器	0x0
+0x04	R	6	RtcMinute	RTC minute register，分計數器	0x0
+0x08	R	5	RtcHour	RTC hour register，時計數器	0x0
+0x0C	R	16	RtcDays	RTC day count register，日計數器	0x0
+0x10	R/W	6	AlarmSecond	RTC second alarm register，秒鬧鈴暫存器	0x3F
+0x14	R/W	6	AlarmMinute	RTC minute alarm register，分鬧鈴暫存器	0x3F

索引位址	R/W	位元寬度	名稱	設定說明	預設值
+0x18	R/W	5	AlarmHour	RTC hour alarm register，時鬧鈴暫存器	0x1F
+0x1C	R/W	32	RtcRecord	RTC record register	0x0
+0x20	R/W	7	RtcCR	RTC control register	0x0
+0x24	R/W	6	WRtcSecond	RTC second counter write port	0x0
+0x28	R/W	6	WRtcMinute	RTC minute counter write port	0x0
+0x2C	R/W	5	WRtcHour	RTC hour counter write port	0x0
+0x30	R/W	16	WRtcDays	RTC day counter write port	0x0
+0x34	R/W	5	IntrState	RTC interrupt register，	0x0
+0x38	R/W	32	RtcDivide	RTC frequency divider	0x0
+0x3C	R	32	RtcRevision	RTC Revision Register	-

表 7.3.2-2 即時時鐘控制暫存器（RTC control register）位元功能設定

位元	位元名稱	功能設定
0	RTC enable	RTC interrupt enable 0：Disable 1：Enable：啟動中斷請求功能
1	RTC interrupt every second	RTC auto alarm per second 0：Disable 1：Enable（每秒送出一次 rtc_sec 中斷）
2	RTC interrupt every minute	RTC auto alarm per minute 0：Disable 1：Enable（每分送出一次 rtc_min 中斷）
3	RTC interrupt every hour	RTC auto alarm per hour 0：Disable 1：Enable（每小時送出一次 rtc_hour 中斷）
4	RTC interrupt every day	RTC auto alarm per day 0：Disable 1：Enable（每天送出一次 rtc_day 中斷）
5	RTC alarm interrupt	RTC alarm interrupt 設為 1 時，只要計時器符合所有鬧鈴暫存器的設定時就發出 rtc_alarm 中斷請求。
6	RTC Counter Load	RTC counter load 寫入 1 時，會將 WRtcSecond、WRtcMinute、WRtcHour、WRtcDays 等暫存器的值分別載入 RtcSecond、RtcMinute、RtcHour、RtcDays。此位元會自動清除為 0，當此位元由 1 回復成 0 時，會將 RtcSecond 計數器加 1。

表 7.3.2-3 中斷狀態暫存器

Bit	Name	R/W	預設初值	Description
0	Rtc_Second	R/W	0x0	表示 rtc_sec 自動鬧鈴中斷發生
1	Rtc_Minute	R/W	0x0	表示 rtc_min 自動鬧鈴中斷發生
2	Rtc_Hour	R/W	0x0	表示 rtc_hour 自動鬧鈴中斷發生
3	Rtc_Days	R/W	0x0	表示 rtc_days 自動鬧鈴中斷發生
4	Rtc_Alarm	R/W	0x0	表示 rtc_alarm 中斷發生，也就是到達鬧鈴設定時間

以上除了尚未說明 FTRTC010 工作時脈設定外，其餘藉由表 7.3.2-1 ~ 表 7.3.2-3 的說明已經可以進行 FTRTC010 有關的使用設定，表 7.3.2-4 就是有關計時器計數來源的設定，原則上經過 RtcDivide 暫存器的設定後必須產生 1Hz 的時脈給秒計時器使用，在 XC5 平台中要決定送到 RTC 的時脈則由電力管理晶片（PMU）的 OSC 控制暫存器設定（OSC Control Register，OSCCR），OSCCR[RTCLSEL]=1 時，會將 32.768KHz 送至 RTC 中，否則送至 5MHz 的時脈

表 7.3.2-4 RtcDivide 暫存器使用列表

Bit	Name	R/W	預設初值	Description
31	DividerEnable	R /W	0x0	1：啟動除頻電路，EXTCLK 經過 DividerCycle 除頻，再送到秒計時器進行計數，否則 EXTCLK 直接做秒計時器的計數時脈。
30:0	DividerCycle	R /W	0x8000	EXTCLK/DividerCycle 的頻率才是計時器使用的計數頻率，基本上應該是 1Hz 的時脈。因此，如果 EXTCLK 採用 32.768KHz 的時脈，則 DividerCycle 就應設為 0x8000。

7.3.3 ADP-XC5-for-N801-S RTC 實作

- **N8_RTC 實作**

本單元旨於利用 RTC 單元提供的即時時鐘的功能模擬出一數位時鐘，將時間透過 UART 傳回開發主機的終端機程式，整個專案只有一個程式碼

rtc.c，計時的初值經由寫入 WRtcXxx 等四個暫存器設定，計時脈波由 XC5 提供週期 1 秒的脈波，對 RtcCR 寫入 0x7F 將 WRtcXxx 等計時初值載入計時計數器（RtcSecond…）中，並啟動計時。此外設定四個變數（times.second…等）用來記錄目前的計時時間，其初值由計時計數器讀取而得，並利用日、時、分、秒的計時中斷來計數，每一秒都透過 UART 傳回開發主機終端機程式，由於 XC5 提供的計時脈波需校正，因此每一秒中斷發生時加上一小段延遲校正。

```
int main(void)
{ Time time;
        RtcDivide = 0x00000000;//不啟動除頻電路
        WRtcSecond = 50; //設定時間初值
        WRtcMinute = 8;
        WRtcHour = 21;
        WRtcDays = 0;
//將 WRtcSecond/WRtcMinute/WRtcHour/WRtcDays 各暫存器的值 Load 至
//RtcSecond/RtcMinute/RtcHour/RtcDays
        RtcCR = 0x0000007F; //致能 RTC 與中斷信號
        while(RtcCR & 0x40) //判斷 RtcCR 暫存器的 Rtc Counter Load 是否變為 0
        {printf("Load set time:%x\n",RtcCR);}
        time.second = RtcSecond; //時、分、秒、日的計時計數暫存器的值存到變數
        time.minute = RtcMinute;
        time.hour = RtcHour;
        time.days = RtcDays;
```

```
        while(1)
        {if(IntrState & 0x01) //Rtc"秒"中斷發生
                {if (IntrState & 0x02) //Rtc"分"中斷發生
                        {if(IntrState & 0x04) //Rtc"時"中斷發生
                                {if(IntrState & 0x08) //Rtc"日"中斷發生
                                        {intrState = 0x0;
                                                time.days++;
                                                if(time.days > 30)
                                                {time.days = 0;} //日期歸零
                                        }
                                        IntrState = 0x0; //清除中斷狀態暫存器
                                        time.hour++;
                                        if(time.hour > 23 )
                                        {time.hour = 0;} //時歸零
                                }
                                IntrState = 0x0; //清除中斷狀態暫存器
                                time.minute++;
                                if(time.minute > 59)
                                {time.minute = 0;  }//分歸零
                        }
                        IntrState = 0x0; //清除中斷狀態暫存器
                        time.second++;
                        if(time.second > 59)
                        {time.second = 0;  } //秒歸零
        printf("%dd:%dh:%dm:%ds\n", time.days, time.hour, time.minute, time.second);
                        delay(400000);//校正 XC5 提供的秒計時脈波
                }
        }
        return EXIT_SUCCESS;
}
```

此專案的實作步驟可以依第四章匯入專案目錄（請參考 N8_PILI）或是建立專案後只匯入程式碼 rtc.c（請參考 N8_7SEG）皆可輕易完成整個專案（N8_RTC 專案）的建置。此專案執行時會透過 UART 傳回日時分秒到開發主機，因此在此專案利用 AndeSight 提供的終端機程式接收時間並顯示，開啟終端機程式視窗方法如圖 7.3.3-1，開啟終端機設定視窗如圖 7.3.3-2 畫面，通訊埠、鮑率、資料位元、停止位元、同位位元等設定如圖 7.3.3-3。終端機連線設定後會自動與 XC5 連線，AndeSight 可以直接透過 AICE 下達

重置 XC 的指令，點選工作平台後（圖 7.3.3-4 ①），按滑鼠右鍵出現下拉式選單以滑鼠左鍵點擊「Reset Target」（圖 7.3.3-4 ②）可重置 XC5 平台，平台重置後透過 UART 傳送訊息（圖 7.3.3-4 ③）至開發主機，圖 7.3.3-4 終端機視窗中可以看見 XC5 傳回的訊息。圖 7.3.3-5 是 N8_RTC 專案執行結果。

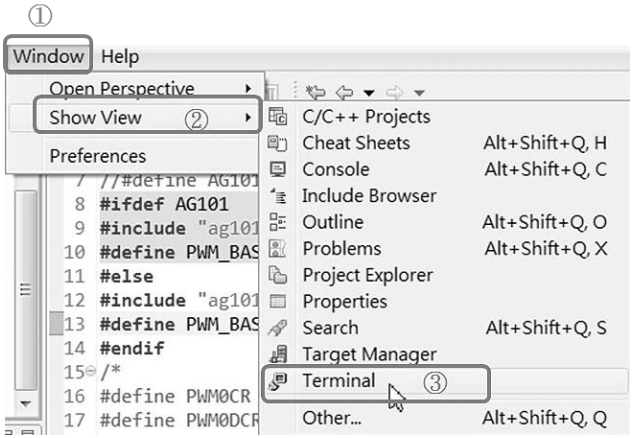

圖 7.3.3-1 開啟終端機程式視窗

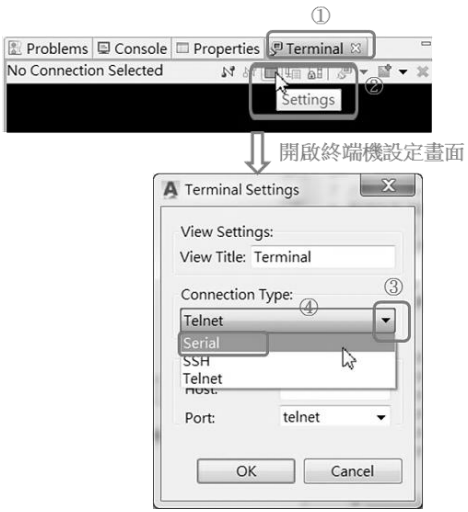

圖 7.3.3-2 開啟終端機設定視窗

圖 7.3.3-3 終端機程式通訊協定設定

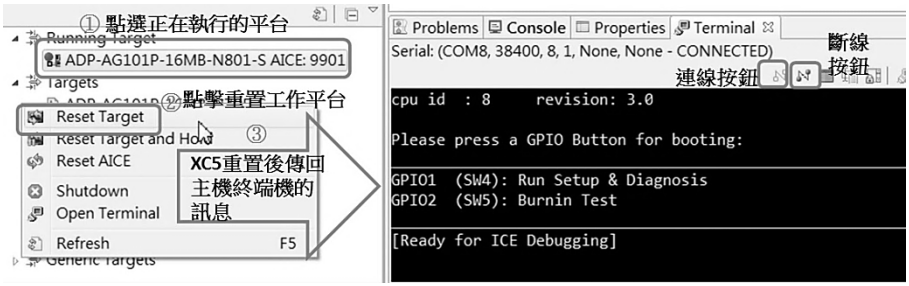

圖 7.3.3-4 AndeSight 終端機接收 XC5 重置後傳回的訊息

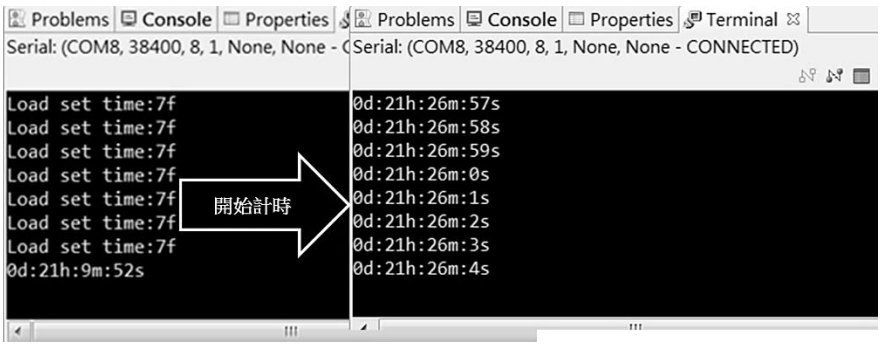

圖 7.3.3-5 N8_RTC 執行畫面