

第五章

萬用非同步串列傳輸

5.1 萬用非同步串列傳輸 (UART)

萬用非同步傳輸是古老但常用的串列傳輸方式，其基本傳輸資料格式如圖 5.1-1 所示，未傳輸資料時傳輸線上呈現高電位 (Mark)，開始傳輸資料時第一個信號是起始位元 (Start Bit) 電氣信號是低電位 (Space)，接著是 UART 接收器據此判斷傳送者開始傳送資料，真正的資料位元以最小權位位元 (LSB) 先傳送，一般 UART 可以設定資料位元為 5 ~ 8 位元，資料位元傳輸完畢後緊接著依使用者設定決定是否傳送同位位元 (Parity) 或奇同位 (Odd) 或偶同位 (Even) 位元，傳輸終了會傳送停止位元 (Stop Bit)，其電氣信號固定是高電位 (Mark)，以上數位信號採用正邏輯為之，但是在 RS232 傳輸協定中是采用負邏輯，換言之起始位元的電氣信號是高電位 (+5 ~ +15V)，停止位元是低電位 (-5 ~ -15V)，在 UART 控制器傳送或接收資料前必須經過電位轉換單元，轉換成負邏輯電氣信號並符合 RS232 的規格，常見的電氣信號轉換元件是 MAX232。

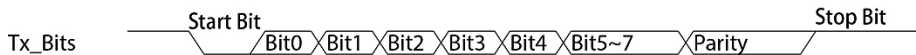


圖 5.1-1 萬用非同步串列傳輸資料格式

5.2 WT59F064 萬用非同步串列傳輸控制器

5.2.1 WT59F064 UART 特性

WT59F064 中共提供 4 組萬用非同步串列傳輸控制器 UART+0 ~ 3，其占用位址空間分別是 0x00203400 ~ 0x002037FF、0x00203800 ~ 0x00203BFF、0x00203C00 ~ 0x00203FFF、UART3 0x00204000 ~ 0x002043FF。此控制器的特性與常見的 16550 UART 接近，萬用非同步串列傳輸控制器的工作原理可以參考 Wiki (<http://goo.gl/FbVgqv>)，在此簡述其特性：

- 全雙工傳輸

- NRZ 編碼方式
- 8 或 9 個資料位元
- 1 停止位元 (Stop bit)
- 可設為無同位、奇同位或偶同位
- 傳送或接收可各別致能啟動
- 提供傳收偵測旗標(傳輸完畢、接收緩衝器已填滿、傳送緩衝器空閒)
- 提供四類錯誤旗標 (含過衝 (Overrun)、資料框 (frame) 錯誤等等)
- 四類中斷源旗標 (同上)
- 多處理器通訊 (未符合位址時進入靜音模式)
- 可於靜音模式喚醒

5.2.2 WT59F064 鮑率 (Baud Rate) 計算方式

WT59F064 萬用串列傳輸的鮑率計算公式如下：

$$Baud\ Rate = \frac{Pclk}{16(Mantisa + \frac{Fraction}{16})}$$

Mantisa 與 *Fraction* 的設定如下所示：

$$Mantisa = \left\lfloor \frac{Pclk}{16 * Baud\ Rate} \right\rfloor$$

$$Fraction = \left\lfloor \left(\frac{Pclk}{16 * Baud\ Rate} - Mantisa \right) * 16 \right\rfloor$$

由於 ADP-WT59F064 系統時脈使用 24Mhz 的時脈，如果傳輸鮑率採用 38400bps 的話，*Mantisa* 就必須設為 39 (0x27)，*Fraction* 設為 1，表 5.2.2-1 列出各種傳輸速率所需的 *Mantisa* 與 *Fraction* 的設定值。

表 5.2.2-1 不同傳輸鮑率所需的設定值 (系統時脈為 24Mhz)

Baud Rate(bps)	Mantisa	Fraction	誤差(%)
2400	625	0	0
4800	312	8	0

Baud Rate(bps)	Mantisa	Fraction	誤差(%)
9600	156	4	0
19200	78	2	0
38400	39	1	0
57600	26	1	0.08
115200	13	0	0.16

5.2.3 WT59F064 UART 暫存器說明與使用設定

在使用 UART 前首要了解各暫存器的物理意義，了解暫存器中各位元的設定方式，依連線傳輸需求設定各暫存器，表 5.2.3-1 列出 UART 中暫存器的名稱與索引位址及其重置 (Reset) 後預設初值，其中控制暫存器 (CR, Control Register) 有 LCR1 主要用來啟動傳輸與接收，並且用來設定傳輸資料位元長度及同位極性位元等；LCR2 主要用來設定中斷功能的啟動與否。讀取狀態暫存器 (SR, Status Register) 則可得知傳輸是否完畢、是否已接收資料、閒置 (Idle) 與斷線 (Break) 是否偵測到、錯誤是否發生等。資料傳送暫存器 (TDR, Transmit Data Register) 與資料接收暫存器 (RDR, Receive Data Register) 則用於傳送與接收資料，欲傳輸資料時將欲傳送的資料位元組寫入 TDR，UART 控制器會將 TDR 中的位元組資料移入移位暫存器然後一個位元一個位元逐一經由傳輸接腳送出；相反地當資料接收腳位將輸入資料逐一透過移位暫存器接收完畢後會將資料位元組送到 RDR，可以透過中斷通知或由主程式主動讀取狀態暫存器得知接收的資料已置於 RDR，讀取 RDR 即可得到接收到的資料。進入實作前必須了解 UART 程式的寫作流程：1. 設定系統控制暫存器 1 (0x200000) 的 bit0 ~ 4 選擇外部 24Mhz 石英震盪器做為系統時脈來源，並透過系統控制暫存器 2 (0x200004) 的 bit0 設定 UART 的工作時脈來源為外部時系統時脈；2. 設定 PORTA_9、PORTA_10 腳位分別為 UART 0 的輸出與接收腳位；3. 依傳輸速率，由表 5.2.2-1 (系統時脈為 24Mhz) 選擇正確數值填入暫存器 BRR 中；4. 依最常見的傳輸模式將 LCR1 設為 0x3800 (8 data bits、non-parity，啟動傳輸與接收功能)，以上完成後即可進行非同步串列傳送與接收。

WT59F064 UART 常用暫存器列表如下：

表 5.2.3-1 WT59F064 UART 暫存器索引位址列表

Reg. Name	索引位址	重置初值
控制暫存器 1 (Control Register 1)(LCR1)	+ 0x0	Bit[13..0]=0
控制暫存器 2 (Control Register 2)(LCR2)	+0x04	Bit[8..0]=0
狀態暫存器 (Status Register)(LSR)	+ 0x08	Bit[4]=0
資料傳送暫存器 (Transmit Data Register)(TDR)	+0x0C	未知
資料接收暫存器 (Receive Data Register)(RDR)	+ 0x10	未知
鮑率設定暫存器 (Baud Rate Register)(BRR)	+0x14	Bit[3..0]=0
斷線設定暫存器 (Break Register)(LBR)	+ 0x18	Bit[3..0]=0

表 5.2.3-2 WT59F064 UART LCR1 暫存器個別位元設定說明

Bit	R/W	Name	Function Description
13	R/W	UE	UART 致能啟動 0：禁止 UART 預除及輸出 1：啟動 UART
12	R/W	TE	Transmitter enable 0：禁止傳送輸出 1：啟動傳送輸出功能
11	R/W	RE	Receiver enable 0：禁止接收 1：啟動接收並於輸入端準備接收起始位元
10	R/W	M	Word length 0：1 起始位元，8 資料位於，n Stop bit 1：1 Start bit, 9 Data bits, n Stop bit
9	R/W	DMAT	DMA enable transmitter 1：傳送時啟動 DMA 傳輸模式 0：傳送時禁止 DMA 傳輸模式
8	R/W	DMAR	DMA enable receiver 1：接收時啟動 DMA 傳輸模式 0：接收時禁止 DMA 傳輸模式
7	R/W	RWU	Receiver wakeup 0：接收器正常啟動 1：使接收器進入喚醒啟動功能
6	R/W	WAKE	Wakeup method (喚醒方式) 0：傳輸線閒置的方式喚醒 1：依符合位址標示的方式喚醒
5:2	R/W	ADD	Address of the USART node 設定 UART 的位址，通常用來喚醒 UART 用
1	R/W	PCE	Parity control enable 0：沒有同位位元 1：有同位位元

0	R/W	PS	Parity selection 0：偶同位 1：奇同位
---	-----	----	------------------------------------

表 5.2.3-3 WT59F064 UART LCR2 暫存器說明

Bit	R/W	Name	Function Description
8	R/W	IDLEIE	IDLE interrupt enable 0：傳輸線閒置時禁止中斷 1：傳輸線閒置時可啟動中斷
7	R/W	BDIE	BD interrupt enable 致能斷線時啟動中斷請求
6	R/W	TXEIE	TXE interrupt enable 0：當 TXE=1 禁止中斷 1：當 TXE=1 時啟動中斷請求
5	R/W	TCIE	Transmission complete interrupt enable 0：Interrupt is inhibited 1：An USART interrupt is generated whenever TC=1
4	R/W	RXNEIE	RXNE interrupt enable 0：Interrupt is inhibited 1：An USART interrupt is generated whenever ORE=1 or RXNE=1
3	R/W	EIE	Error Interrupt enable 0：Interrupt is inhibit 1：An interrupt is g or ORE=1 or NE=1
0	R/W	PEIE	PE interrupt enable 0：Interrupt is inhibited 1：PE interrupt enable

表 5.2.3-4 WT59F064 UART LSR 暫存器說明

Bit	R/W	Name	Function Description
8	R	IDLE	IDLE interrupt enable 0：Interrupt is inhibited 1：An USART interrupt is generated whenever IDLE=1
7	R	BD	BD interrupt enable
6	R/W	TXE	TXE interrupt enable 0：Interrupt is inhibited 1：An USART interrupt is generated whenever TXE=1
5	R	TC	Transmission complete interrupt enable 0：Interrupt is inhibited 1：An USART interrupt is generated whenever TC=1
4	R/W	RXNE	RXNE interrupt enable 0：Interrupt is inhibited

			1 : An USART interrupt is generated whenever ORE=1 or RXNE=1
3	R	ORE	Overrun error 1 : 表示 MCU 來不及讀取接收到的字元。導致接收移位暫存器的資料無法送進資料接收暫存器而被後來的接收字元覆蓋。讀取資料接收暫存器將可清除此字元。
2	R	NE	Noise error flag 1 : 表示偵測到雜訊。
1	R	FE	Frame error 1 : 表示接收到斷線字元或是接收到錯誤資料格式（無法正確接收到停止字元）
0	R	PE	Parity Error 0 : 無同位錯誤。 1 : 表示同位位元錯誤。

表 5.2.3-5 WT59F064 UART TDR 暫存器說明

Bit	R/W	Name	Function Description
8:0	R/W	TDR	Transmit data value

表 5.2.3-6 WT59F064 UART RDR 暫存器說明

Bit	R/W	Name	Function Description
8:0	R/W	RDR	Received data value

表 5.2.3-7 WT59F064 UART BRR 暫存器說明

Bit	R/W	Name	Function Description
15:4	R/W	MANTISSA	mantissa of baud rate generator
3:0	R/W	FRACTION	fraction of USARTDIV

5.2.4 ADP-WT59F064-UART 實作

此實驗旨在練習非同步萬用串列傳輸，透過 WT59F064 UART0 傳送資料以 RS232 的電器信號送出，開發主機端透過終端機程式或 AndeSight IDE 提供的終端機程式接收並顯示。開發主機端終端機程式需先設定與 WT59F064 UART0 一致的傳輸速率與資料格式，目前程式中設定鮑率 38400 bps、8 個資料位元、無同位位元及 1 個停止位元。整個專案中 main.c 是主要程式，UART0 的設定與傳送皆在此程式完成，main() 中首先透過 system_control_00 與 system_control_04 兩變數選擇系統使用 24MHz 的時脈來源

»» 微處理器應用與實作：C 語言與 Andes MCU 系列

做為系統時脈 [System_Control_00=(System_Control_00 | 0x000E)、System_Control_04=(System_Control_04 | 0x0001)]。接著呼叫 DRV_UartInitial () 進行 UART0 的硬體初始化設定，初始化的過程先將 PORTA 9 與 10 設為特殊功能供 UART 使用 (GPIO_ACT_PA=0xF9FF)，接著設定傳輸資料格式與速率 [UART0_LCR=0x3801 ; UART0_BAUD=0x271]，UART0 初始化完畢後再來就是進行 LCM 的初始化及顯示字串「UART0 Transfers Digits」，以上工作準備完畢後開始透過 DRV_PutStr() 函式，將置於陣列變數 buf[] 的字串傳送出去，並將其顯示於 LCM。以下是匯入程式及進行終端機設定的步驟：

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」做專案匯入 UART。

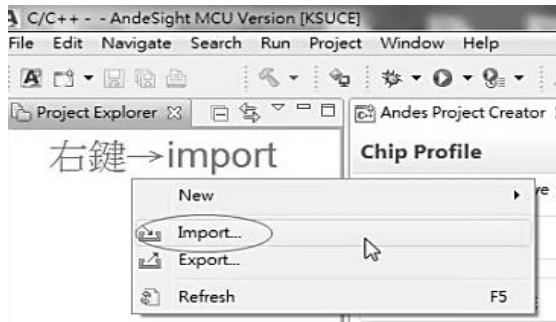


圖 5.2.4-1 開啟匯入 UART 專案視窗

Step 2. 圖 5.2.4-1 中點擊「General」左方三角標誌開啟選項，選擇「Existing Projects into Workspace」後 (圖 5.2.4-1 ①)，按 Next > (圖 5.2.4-1 ②)。

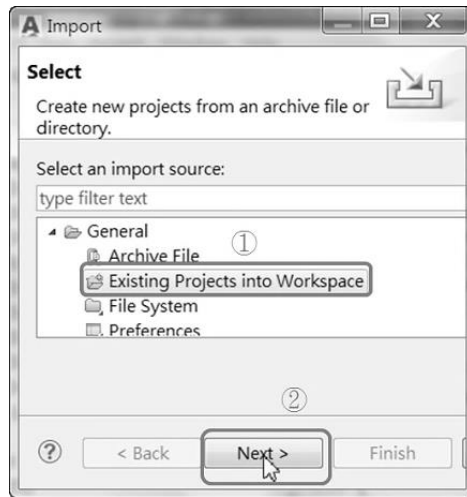


圖 5.2.4-2 匯入既有的 UART 專案

Step 3. 因為匯入的是專案壓縮檔因此點擊「Select archive file」，(圖 5.2.4-3 ①) 後再點擊「Browse」(圖 5.2.4-3 ②) 選項選擇要匯入的專案。

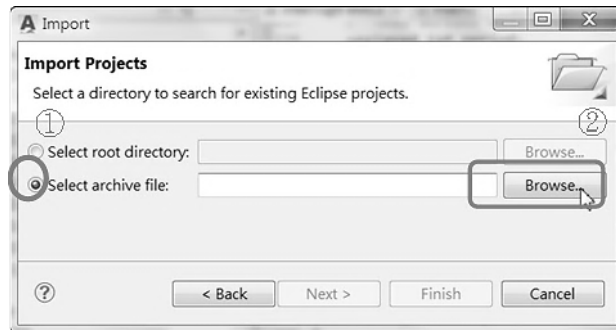


圖 5.2.4-3 打開瀏覽工具選擇正確的專案檔

Step 4. 點選「ADP-WT59F064-UART.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-UART)，在專案檔名稱左方勾選 (圖 5.2.4-4 ①) 表示要匯入此專案，然後按點擊下方「Finish」(圖 5.2.4-4 ②) 選項。

»» 微處理器應用與實作：C 語言與 Andes MCU 系列

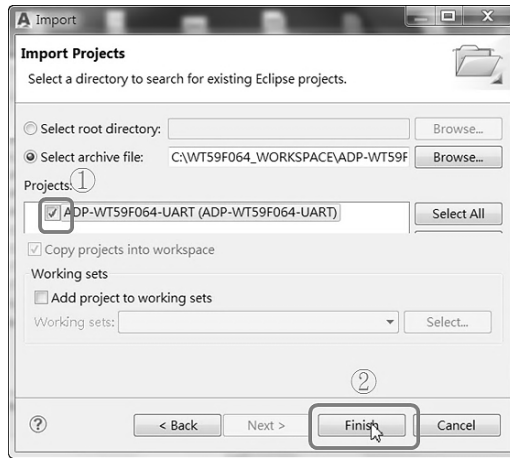


圖 5.2.4-4 選擇 ADP-WT59F064-UART 完成專案匯入

Step 6. 透過「Build Project」編譯專案檔裡的程式碼。先以滑鼠左鍵點選專案「ADP-WT59F064-UART」(圖 5.2.4-5 ①)使其出現反白然後按滑鼠右鍵出現下拉式選單後，點選「Build Project」(圖 5.2.4-5 ②)進行編譯連結的工作。

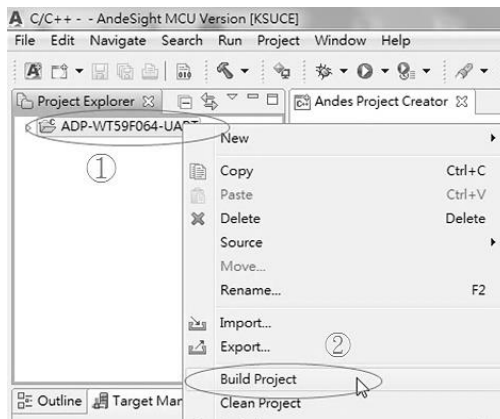


圖 5.2.4-5 編譯連結程式

Step 7. 點擊工具列上「Window」選項(圖 5.2.4-6 ①)出現下拉式選單，游標移至「Show View」選項(圖 5.2.4-6 ②)時會再出現下一層選單，以滑鼠左鍵點選「Terminal」選項(圖 5.2.4-6 ③)就會出現終

端機程式視窗（圖 5.2.4-7）。因為此專案執行時會透過 J9（RS232 埠）將字串傳回到開發主機，因此利用 AndeSight V2.0 之後版本提供的終端機程式來接收傳回的字串，以檢驗程式執行是否正確。

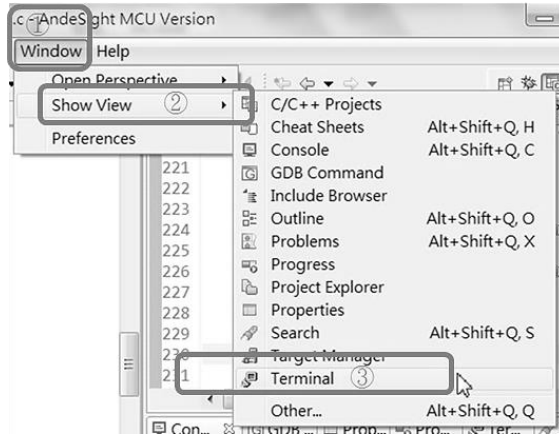


圖 5.2.4-6 開啟 Terminal 終端機視窗

Step 8. 設定終端機程式使用的 COM 埠、鮑率、資料格式（8 位元、無同位元、1 個停止位元）、無流量控制（flow control）等等。首先以滑鼠左鍵點擊圖 5.2.4-7 ① 的設定符號，開啟終端機程式設定畫面並依序選 COM 埠、鮑率（38400bps），8 個資料位元、1 個停止位元、無同位元（圖 5.2.4-7 ②），再點擊「OK」（圖 5.2.4-7 ③）。

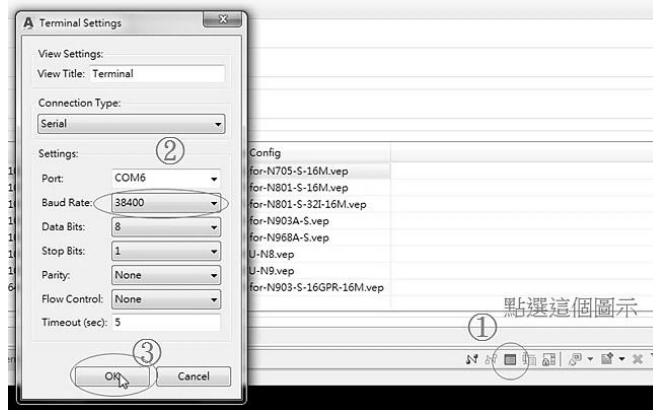


圖 5.2.4-7 AndeSight 終端機程式設定

»» 微處理器應用與實作：C 語言與 Andes MCU 系列

Step 9. 把 Step 6. 中建置完成的「ADP-WT59F064-UART.bin」檔透過燒錄程式 ISP_WT59F064.exe 燒錄到 WT59F064 晶片中。對 ADP-WT59F064-UART 按右鍵出現下拉式選單（圖 5.2.4-8 ①），接著點擊「Flash Burner」（圖 5.2.4-8 ②）以開啟燒錄視窗。

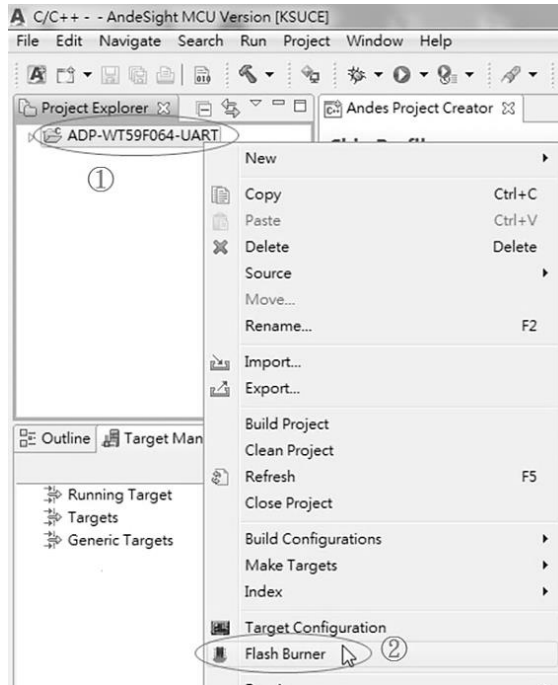


圖 5.2.4-8 開啟燒錄程式

Step 10. 燒錄程式視窗中可以見到燒錄程式碼檔名，要燒錄到晶片中的程式碼路徑及檔案名稱，實際上燒錄前燒錄程式會先檢查 flash memory 是否已經清空，清空後再燒錄，燒錄後會再檢查燒錄到晶片的程式碼與存在檔案中的程式碼是否一致，這些步驟使用者可以分開來執行，或是透過點擊「Auto」一次完成，整個燒錄過程如圖 5.1-1 所示，當出現「Verify Successful」，表示燒錄成功。程式執行畫面如圖 5.2.4-10 下方終端機程式裡出現的字串就是 ADP-WT59F064 透過 UART 0 傳送出來的字串。



圖 5.2.4-9 程式燒錄工具視窗



圖 5.2.4-10 程式碼燒錄過程

5.3 ADP-XC5-for-N801-S 萬用非同步串列傳輸控制器

5.3.1 ADP-XC5-for-N801-S UART 特性

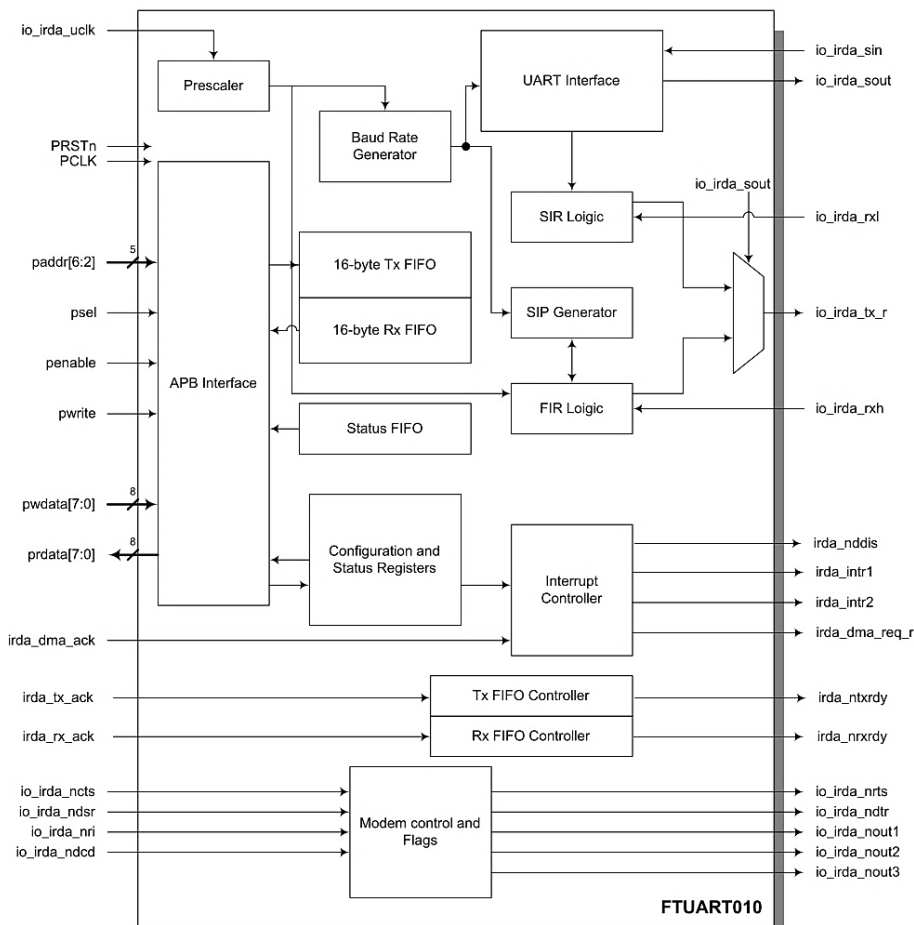


圖 5.3.1-1 N8 SoC UART 方塊圖

ADP-XC5-for-N801-S 提供 2 個萬用非同步串列傳輸控制器，一個是 BTUART (BlueTooth UART) (位址：0xF03000)，一個是 STUART (Stan

dard UART)(位址：0xF16000)，使用上並無太大差異只是前者多了兩個調變與解調器 (Modem) 用的控制信號 nCTS 與 nRTS，若需要其他相關控制信號可利用 GPIO 模擬提供。此 UART 特點有：

- 完全相容高速 UART NS16C550A
- 鮑率可達 115.2Kbps，可規劃傳輸鮑率
- 傳輸字元可含 5 ~ 8 位元，可規劃成無極性位元傳輸或同位位元傳輸，可設定為 1、1.5 或 2 位元的停止位元。
- 可讀取傳輸狀態或錯誤狀態所有訊息
- 可主動斷線或偵測斷線狀況
- 具有優先中斷控制
- 支援 DMA 模式傳輸與接收
- 提供錯誤模擬 (Parity、Break、overrun、framing error)
- BTUART 提供 32 位元組的先進先出 (FIFO) 接收與傳送緩衝器
- STUART 提供 16 位元組的先進先出 (FIFO) 接收與傳送緩衝器

5.3.2 N801 系統晶片 UART 鮑率設定

N801 系統晶片提供的 UART 包含紅外線接收器的功能，在此僅介紹一般串列傳輸功能，表 5.3.2-1 僅列出一般萬用非同步串列傳輸控制器常用的暫存器名稱與索引位址，DLAB 位元位於 LCR 的 Bit 7，當要設定傳輸鮑率時要先設定此位元，才能將除頻值放入 DLL 與 DLM 中得到正確的傳輸速率，當要讀取接收資料或者是傳送資料時須將 DLAB 設定為 1。由於 ADP-XC5-for-N801-S 系統周邊匯流排時脈為 15Mhz，因此除頻值的設定如表 5.3.2-2 所列。

表 5.3.2-1 ADP-XC5-for-N801-S UART 列表 (不含 FIR 工作模式)

索引位址	R/W	名稱	預設值
DLAB=0			
+0x00	R	接收暫存器Receiver Buffer Register(RBR)	0x00
	W	傳送暫存器Transmitter Holding Register(THR)	0x00
+0x04	R/W	中斷致能暫存器Interrupt Enable Register(IER)	0x00
+0x08	R	中斷識別暫存器Interrupt Identification Register(IIR)	0x01
	W	先進先出緩衝控制暫存器FIFO Control Register(FCR)	0x00
+0x0C	R/W	傳輸線控制暫存器Line Control Register(LCR)	0x00
+0x10	R/W	調變解調變控制暫存器Modem Control Register(MCR)	0x00
+0x14	R	傳輸線狀態暫存器Line Status Register(LSR)	0x60
	W	測試暫存器Testing Register(TST)	0x00
+0x18	R	調變解調變狀態暫存器Modem Status Register (MSR)	0x00
+0x1C	R/W	解析暫存器Scratch Pad Register(SPR)	0x00
DLAB=1			
+0x00	R/W	低位元組鮑率除頻暫存器Baud Rate Divisor Latch Least Significant Byte(DLL)	0x01
+0x04	R/W	高位元組鮑率除頻暫存器Baud Rate Divisor Latch Most Significant Byte(DLM)	0x00
+0x08	R/W	預除暫存器Prescaler Register(PSR)	0x01

表 5.3.2-2 N8 SoC UART 鮑率與除頻值對應值

Baud Rate(bps)	DLM	DLL	誤差(%)
2400	1	135	0.096
4800	0	195	0.16
9600	0	98	0.35
19200	0	49	0.35
38400	0	24	1.725
57600	0	16	0.08
115200	0	8	0.16

5.3.3 ADP-XC5-for-N801-S UART 暫存器說明與使用設定

- 資料傳送與接收緩衝暫存器 (RBR&THR，索引位址 0x0)

要進行資料傳輸前必須先將 DLAB 設為 0 (LCR[7]=0)，雖說兩個暫存器位址一樣但是硬體藉由讀寫控制信號區別是使用哪個暫存器。由於此

UART 具有 FIFO 緩衝器因此當啟動 FIFO 功能時，資料的傳輸或接收就各別有兩個路徑，一個是經由 FIFO 緩衝器傳送或接收，當 FCR[0]=1 時啟動 FIFO 功能，使用者欲傳送的資料雖說是寫入索引位址為 0 的暫存器，但實際是寫入 FIFO 輸出緩衝器中，並且由 FIFO 輸出緩衝器提供輸出資料；相反地 FCR[0]=0 時使用者欲傳送的資料會送入 FIFO 與 THR，但實際送出資料的是經由 THR 暫存器送出，FCR[0]=0 時所讀取的接收資料位元組是由 RBR 讀取。

表 5.3.3-1 傳送 / 接收緩衝暫存器

索引位址	R/W	名稱	位元數	預設值
+0x00	R	接收暫存器Receiver Buffer Register(RBR)	8	0x00
	W	傳送暫存器Transmitter Holding Register(THR)	8	0x00

● 傳輸線控制暫存器 (LCR , 索引位址 0x0C)

此暫存器主要設定傳輸資料格式包括字元長度、停止位元數、同位位元等，此外可將傳輸線設為斷線 (logic 0)。設定方式如表 5.3.3-2 中功能欄中所述。

表 5.3.3-2 傳輸線控制暫存器 LCR (索引位址 0x0C)

Bit	Name	Type	Function
7	DLAB	R/W	Divisor Latch Access Bit (DLAB). DLAB=1 時才能讀寫 DLL、DLM、PSR 等暫存器。
6	Set Break	R/W	此位元設定為 1 時，輸出線將成 logic 0(Space)的電器狀態
5	Stick Parity	R/W	LCR[5..3]=0b111 時，同位元會被傳送出去且一定是 0，LCR[5..3]=0b0 時，同位元會被傳送出去且一定是 1。
4	Even Parity	R/W	LCR[4,3]=0b01 時，傳送資料時加上奇同位位元，LCR[4,3]=0b11 時，傳送資料時加上偶同位位元。
3	Parity Enable	R/W	LCR[5]=1 啟動同位元傳輸，再傳送資料時資料位元傳送完畢後在停止位元傳送前加上同位位元。
2	Stop Bits	R/W	LCR[2]=0, 1 個停止位元，LCR[2]=1, 2 個停止位元或 1.5 個停止位元 (5 個資料位元時)。
1	WL1	R/W	[WL1,WL0]. =0b00(字元長度= 5)，[WL1,WL0]. =0b01 (字元長度= 6)，

»» 微處理器應用與實作：C 語言與 Andes MCU 系列

Bit	Name	Type	Function
			[WL1,WL0].:=0b10(字元長度=7), [WL1,WL0].:=0b11 (字元長度=8)。
0	WL0	R/W	同上

● 先進先出控制暫存器 FIFO Control Register (FCR , 索引位址 0x08)

要使用此暫存器必須先將 DLAB 清除為 1，否則此位址為預除暫存器所用。其使用設定如表 5.3.3-3 功能欄的說明。

表 5.3.3-3 FIFO 控制暫存器

Bit	Name	Type	Function
7:6	RXFIFO_TRGL	W	當接收 FIFO 中的資料個數到達此一臨界設定值即產生中斷信號。
5:4	TXFIFO_TRGL	W	當傳送 FIFO 緩衝器中的資料個數到達此一臨界設定值即產生中斷信號。
3	DMA Mode	W	啟動 DMA 傳輸模式
2	Tx FIFO Reset	W	將 1 寫入此位元將清除傳送 FIFO 緩衝器中所有資料，但不影響正在傳輸的資料。
1	Rx FIFO Reset	W	將 1 寫入此位元將清除接收 FIFO 緩衝器中所有資料，但不影響正在接收的資料。
0	FIFO Enable	W	啟動 FIFO 傳輸模式，同時會清除 FIFO 中的資料。

表 5.3.3-4 接收 FIFO 緩衝器臨界值

FCR Code		16-Byte Receiver's FIFO Trigger Level	32-Byte Receiver's FIFO Trigger Level
Bit 7	Bit 6		
0	0	1字元 (character)	1字元
0	1	4字元	8字元
1	0	8字元	16字元
1	1	14字元	28字元

表 5.3.3-5 傳送 FIFO 緩衝器臨界值

FCR Code		16-Byte Receiver's FIFO Trigger Level	32-Byte Receiver's FIFO Trigger Level
Bit 7	Bit 6		
0	0	1字元 (character)	1字元
0	1	3字元	8字元
1	0	9字元	16字元
1	1	13字元	28字元

- 預除暫存器 Prescaler Register (PSR , 索引|位址 0x08)

欲使用此暫存器必須先設定 DLAB 為 1，此暫存器有 PSR[4..0] 5 個位元，傳輸速率的設定是 $\text{io_irda_uclk} / \text{PSR} / 16$ 而得。

- 中斷致能暫存器 Interrupt Enable Register (IER , 索引|位址 0x0 0x04)

此暫存器可設定哪種狀況發生時可產生中斷請求信號。

表 5.3.3-6 中斷致能暫存器功能設定

Bit	Name	Type	Function
3	MODEM Status	R/W	設定 1 時致能調變解調變器的狀態信號產生中斷的功能
2	Receiver Line Status	R/W	設定 1 時致能接收線傳輸狀態信號產生中斷的功能
1	THR Empty	R/W	設定 1 時致能 THR 空時產生中斷
0	Receiver Data Available	R/W	設定 1 時致能接收到資料或過久沒接收到資料 (FIFO 模式) 時產生中斷

- 中斷來源識別暫存器 Interrupt Identification Register (IIR , 索引|位址 0x08)

當 UART 產生中斷請求信號時，此暫存器可提供產生中斷來源識別，UART 產生的中斷有優先等級之分，其優先順序由高至低順序如下：1. 接收線傳輸狀態 (Receiver Line Status)；2. 接收到資料或過久沒接收到資料

»» 微處理器應用與實作：C 語言與 Andes MCU 系列

料(Receiver Data Available) ; 3. THR 空白時(THR Empty) ; 4. MODEM Status 。

表 5.3.3-7 中斷來源識別暫存器

Bit	Name	Type	Function
7:6	FIFO mode enable	R	FCR [0]=1 時此兩位被設定為 1
5	Reserve	R	These two bits of the IIR are always logic 0.
4	TxFIFO full	R	當傳送 FIFO 滿載時，此位元被設定為 1.
3	FIFO mode only	R	在 FIFO 模式下，當過久沒收到資料時被設定為 1
2:1	Interrupt Identification Code	R	此兩位元代表尚待處理的最高優先次序中斷代號
0	Interrupt Pending	R	0：表示有中斷發生尚未處理 1：表示沒有待處理的中斷

表 5.3.3-8 中斷來源識別及重置

Bit3	Bit2	Bit1	Bit0	優先次序	中斷型態	中斷來源	中斷來源重置
0	0	0	1	----	None	沒有待處理的中斷	None
0	1	1	0	最高優先	Receiver Line Status	可能的中斷原因包含：1. overrun error, 2. parity error, 3. framing error 4. break interrupt	讀取 LSR
0	1	0	0	次高優先	Received Data Ready	非 FIFO 模式：表示 RHR 中存有接收到的資料 FIFO 模式：表示 RX FIFO 緩衝器現有資料個數大於等於接收臨界值。 唯有 RX FIFO 緩衝器中的資料個數小於接收臨界值時此中斷信號才會消除。	讀取 RBR.
1	1	0	0	次高優先	Character Reception Timeout	RX FIFO 中至少有一個字元且超過 4 個字元時間（時間長短依鮑率而定）沒有繼續接收新字元。	讀取 RBR
0	0	1	0	次低優先	沒資料可傳送	非 FIFO 模式情形下，THR 是空的 FIFO 模式情形下，TX FIFO 完全是空的。	寫入資料到 THR 或是讀取 IIR。
0	0	0	0	最低優先	Modem Status	MSR[0]~MSR[3]至少有一位元為 1	讀取 MSR

● 傳輸狀態暫存器 Line Status Register (LSR , 索引|位址=0x14)

此暫存器記錄傳輸時發生的狀況，因此在讀取已收到的字元前需先讀取此暫存器以了解傳輸狀態，否則在讀取資料字元後可能會失去某些訊息。每一位元物理意義說明如表 5.3.3-9。

表 5.3.3-9 傳輸狀態暫存器位元說明

Bit	Name	Type	Function
7	FIFO Data Error	R	非 FIFO 模式：此位元為 0。 FIFO 模式：只要 RX FIFO 發生 parity or framing error 或產生斷線此位元會被設定為 1。 讀取 LSR 及不再有以上 3 種情形發生此位元會被清除。
6	Transmitter Empty	R	當 THR (或 TX FIFO) 且 TSR (傳輸位移暫存器) 沒有待傳輸資料時，此位元被設為 1 表傳輸線上沒有任何傳輸動作。只要將資料寫入 THR 此位元將被清除。
5	THR Empty	R	非 FIFO 模式：THR 空時，LSR[5]=1。 FIFO 模式：TX FIFO 空時，LSR[5]=1 此位元被設為 1 表是 UART 可以接收新資料，即使傳輸線上還在傳送資料位元。當 IER[1]=1 時，此位元若為 1，將發出中斷請求信號。
4	Break Interrupt	R	當接收腳位出現低電位的時間超過一個字元的時間長度，此位元被設為 1。
3	Framing Error	R	當此位元被設定為 1 時表示沒正確接收到停止位元。要重至此位元就需讀取 LSR。
2	Parity Error	R	當當同位位元發生錯誤將此位元設為 1，要重置此位元則需讀取 LSR。
1	Overrun Error	R	當 RSR (接收移位暫存器) 已完整接收一個字元，但是 RBR 有資料等讀取或 FIFO 已滿，此位元皆會被設為 1。此位元皆被設為 1 時，非 FIFO 模式下，在 RBR 的資料不會被覆蓋，但是在 FIFO 模式下，RSR 中的資料將覆蓋 FIFO 中的資料。 讀取 LSR 可清除此位元。
0	Data Ready	R	只要有資料等待讀取此位元被設為 1，所有待讀取資料被讀取完畢才會應除此位元。

»» 微處理器應用與實作：C 語言與 Andes MCU 系列

- 調變與解調變控制暫存器 Modem Control Register (MCR , 索引位址 =0x10)

透過此暫存器的設定與清除可產生 Modem 的控制信號。

表 5.3.3-10 調變與解調變控制器

Bit	Name	Type	Function
6	Out3	R/W	使用直接設定電位
5	DMAmode2	R/W	用以選擇 DMA 傳輸模式
4	Loop	R/W	用以將 UART 設定為迴圈傳輸模式，意即 THR 的資料直接傳送 RBR，未經由輸出接腳輸出
3	Out2	R/W	與 bit 7 相同功能
2	Out1	R/W	與 bit 7 相同功能
1	RTS (Request to Send)	R/W	設定此位元將使得 RTS 腳位送出低電位，用來通知 MODEM 要求輸出資料
0	DTR (Data Terminal Ready)	R/W	此位元用來通知 MODEM 終端機端已經上電準備開始傳輸資料，低準位輸出。

- 調變解調變狀態暫存器 Modem Status Register(MSR, 索引位址=0x18)

表 5.3.3-11 調變解調變狀態暫存器

Bit	Name`	Type	Function
7	DCD	R	Data Carrier Detect (DCD)，腳位 the io_irda_ndcd 輸入的補數，表示偵測到資料載波。
6	RI	R	Ring Indicator (RI)，表示偵測到鈴聲
5	DSR	R	Data Set Ready (DSR)，表示 MODEM 已準備開始進行資料傳輸。
4	CTS	R	回應終端機可以開始傳送資料到 MODEM
3	Delta DCD	R	The delta-DCD flag. 表示自從最後一次讀取 MSR 後，DCD 腳位信號已改變。
2	Trailing edge RI	R	當振鈴信號接腳由低電位變成高電位，此位元被設定為 1
1	Delta DSR	R	表示自從最後一次讀取 MSR 後，DSR 位信號已改變。
0	Delta CTS	R	表示自從最後一次讀取 MSR 後，CTS 位信號已改變。

5.3.4 ADP-XC5-for-N801-S UART 實作

● N8_UART_VEP 實作

本實作旨在熟悉 N801-S UART 的使用與設定，透過 UART 傳送資料，本實作將於 VEP (Virtual Emulation Platform) 完成，利用 AndeSight 系統提供的 SID 中 UART 元件傳輸與顯示資料，實際上 SID UART 元件以連接一個終端機的形式呈現，因此送出的資料會以字元的方式顯示在虛擬終端上。專案模組中的主要程式是 main.c 與 uartio.c。主程式 main() 函式一開始就呼叫 uart_init() 對 UART 進行硬體初始化。

```
void uart_init (void)
{
    //設定鮑率前先設定DLAB
    outw (UART_BASE + 0x0c, inw(UART_BASE + 0x0c) | 0x80);
    outw (UART_BASE + 0x00, ((0x18 >> 0) & 0xff));// 設定 DLL=24
    outw (UART_BASE + 0x04, ((0x18 >> 8) & 0xff));// 設定 DLM=0
    //清除DLAB並設資料傳輸協定8資料位元,1個停止位元,沒有極性位元
    outw (UART_BASE + 0x0c, 0x03);
    //啟動FIFO緩衝功能,清除緩衝器,緩衝器觸發準位設為1個byte
    outw (UART_BASE + 0x08, 0x07);
}
```

uart_init() 設定鮑率前必須先設定 DLAB[outw (UART_BASE + 0x0c, inw(UART_BASE + 0x0c) | 0x80)]，然後才設定除頻值，因為鮑率設為 38400bps，因此除頻值=15000000/16/38400 大約等於 24，所以 DLL=24，DLM=0，[outw (UART_BASE + 0x00, ((0x18 >> 0) & 0xff)); outw (UART_BASE + 0x04, ((0x18 >> 8) & 0xff));]，接著記得清除 DLAB 同時設定資料傳輸格式 [outw (UART_BASE + 0x0c, 0x03);] 為 8 個資料位元、1 個停止位元、無極性位元。

當 UART 初始化完畢後，接著透過 uart_printf() 傳輸字串，在 main() 中以 while() 迴圈重複傳送「Hello World!, i=XXXX」，其中 XXXX 的值由變數「i」提供，該變數每傳送一次字串加 1，作為字串送次數的記錄。實

»» 微處理器應用與實作：C 語言與 Andes MCU 系列

實際上 `uart_printf()` 的執行是呼叫 `vsnprintf()` 將要列印的資料轉換成字串放入陣列緩衝變數 `line_buf[]` 中，再呼叫 `uart_putc()` 將 `line_buf[]` 中的元素以字元為單位呼叫 `uart_outbyte()` 逐一送至 UART 中的輸出緩衝暫存器，交由 UART 傳送移位暫存器一個一個位元送出。當送出換行字元（`0x0a`）時，`uart_putc()` 會執行 `uart_outbyte(0x0d)` 送出游標歸零的動作。由此可見 `uart_outbyte()` 才是直接驅動 UART 的程式碼。

```
void uart_printf(char *format, ...)
{
    #define BUFFER_SIZE 256
    char line_buffer[BUFFER_SIZE];    // output buffer
    unsigned int i = 0;
    va_list args;
    va_start(args, format); // 找到第一個參數
    /* 將輸出資料轉換成 ASCII 字串格式存放於 line_buffer[] */
    vsnprintf(line_buffer, BUFFER_SIZE, format, args);
    va_end(args);
    while (line_buffer[i] != '\0') // 檢查是否已到字串盡頭，如果不是繼續
    {
        uart_putc(line_buffer[i]); // 呼叫 uart_putc() 傳送字元
        i++; // 指向下一個字元
    }
    return;
}

void uart_putc(register unsigned char c)
{
    uart_outbyte(c); // 呼叫 uart_outbyte() 傳送字元
    if (c == 10)
        uart_outbyte((char) 13); // 如果傳送換行字元就補上游標歸位的字元
}
```

```
void uart_outbyte(unsigned char c)
{
    // 測試輸出緩衝器是否已滿，若已滿則重複測試直到有緩衝空間
    while (inw(UART_BASE + 0x08) & 0x10);
    // 緩衝器有空間於是傳送至輸出 FIFO 緩衝器。
    outw (UART_BASE + 0x0, c);
}
```


以下是本專案的執行步驟：

Step 1. 滑鼠點選 File>Import 來載入已建立好的 UART 專案 (圖 5.3.4-1)。

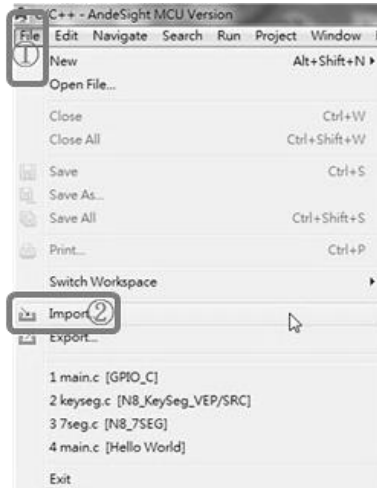


圖 5.3.4-1 匯入專案

Step 2. Select an import source>General>Existing Projects into Workspace，之後按 Next (圖 5.3.4-2)。

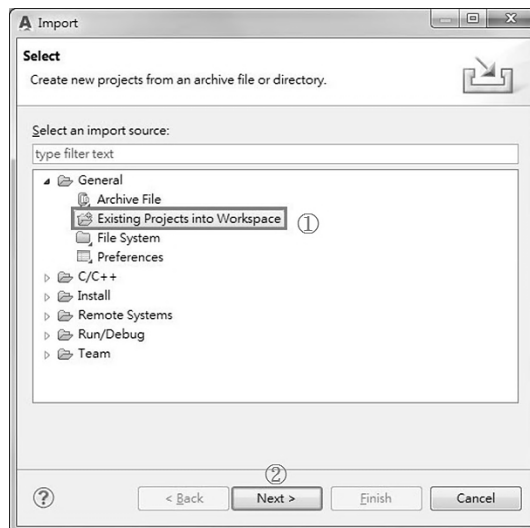


圖 5.3.4-2 選擇開啟既有專案

»» 微處理器應用與實作：C 語言與 Andes MCU 系列

Step 3. 點擊「Browse」(圖 5.3.4-3) 以瀏覽檔案選擇專案目錄。

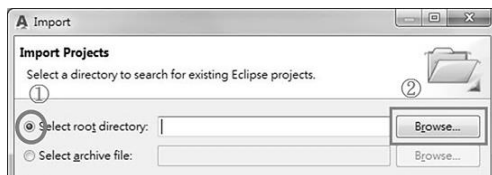


圖 5.3.4-3 點擊「Browse」選擇專案目錄。

Step 4. 該實習專案位置在 N8_WorkSpace>N8_UART_VEP，點選並按確定 (圖 5.3.4-4)。



圖 5.3.4-4 選擇專案目錄

Step 5. 在「Projects」下面可以看到載入的專案名稱及路徑。滑鼠左鍵點擊「Finish」(圖 5.3.4-5) 完成專案匯入。

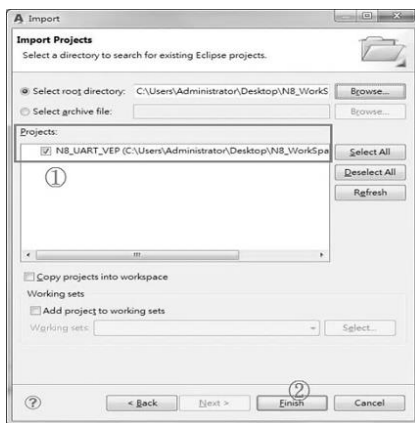


圖 5.3.4-5 選定實際匯入的專案

Step 6. 在「Project Explorer」(圖 5.3.4-6 ①) 可以看到「N8_UART_VEP」專案已載入。滑鼠左鍵點選「N8_UART_VEP」專案後按滑鼠右鍵，點選「Build Project」(圖 5.3.4-6 ②) 開始編譯連結。

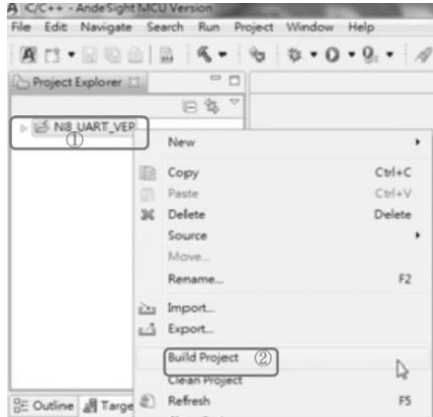


圖 5.3.4-6 開始編譯專案

Step 7. 滑鼠左鍵點擊「N8_UART_VEP」專案後按滑鼠右鍵出現下拉式選單，滑鼠游標移至「Run As」出現下一層選單點擊「(DSF) C/C++ Application」。一般情況下圖 5.3.4-8 不會出現，若出現表示專案中有另外的可執行檔存在，需在圖 5.3.4-8 畫面中點選正確的執行檔再按「OK」開始執行。

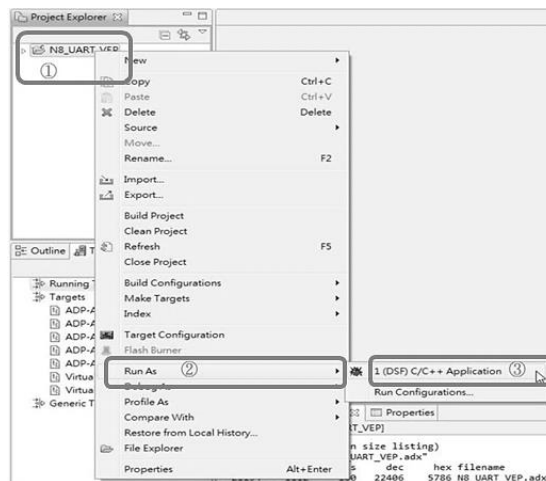


圖 5.3.4-7 準備執行

»» 微處理器應用與實作：C 語言與 Andes MCU 系列

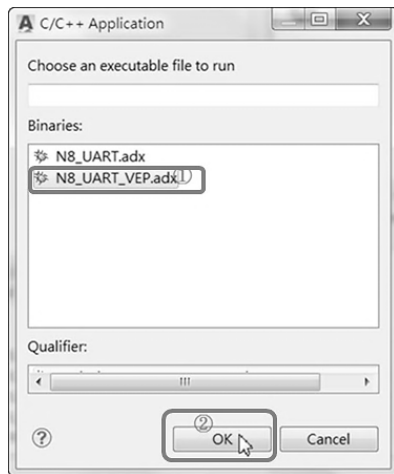


圖 5.3.4-8 選擇正確執行檔

除了以上的執行方式外，可依圖 5.3.4-7 以類似的方式選擇「Run Configurations」(圖 5.3.4-9 ③)，依序在圖 5.3.4-10 按「New」新增一執行設定檔，點擊「New」後出現圖 5.3.4-11，正常情況下會自動填入專案名稱與可執行檔名稱，使用者可以自行透過點選該圖 ①、② 兩項自行選擇專案與可執行檔，由於該專案現在有兩個可執行檔因此直接點選 ② (圖 5.3.4-11) 進行選擇執行檔，此時會出現與圖 5.3.4-8 一致的視窗，依照上述說明進行選擇，選擇完按「OK」後就會將點選的執行檔檔名填入空白欄位上 (圖 5.3.4-12)。最後點擊「Apply」(圖 5.3.4-12 ①) 及「Run」(圖 5.3.4-12 ②) 開始執行程式。

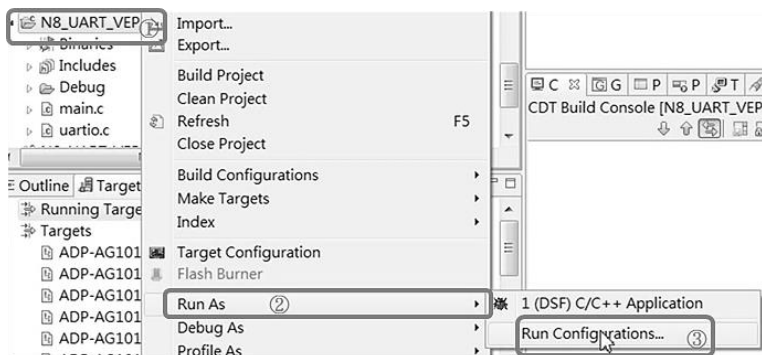


圖 5.3.4-9 執行設定

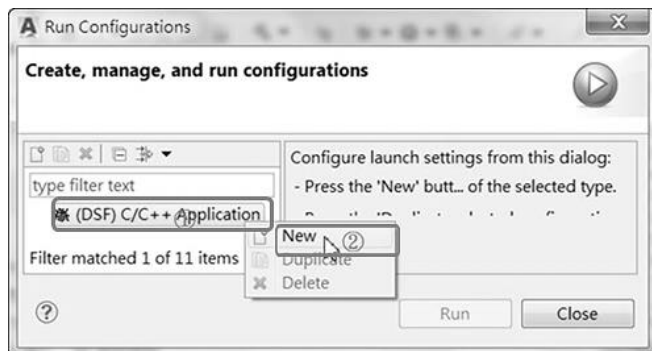


圖 5.3.4-10 新增執行設定檔

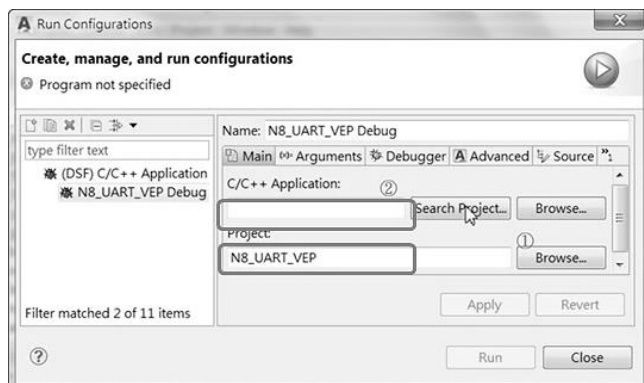


圖 5.3.4-11 選擇專案名稱與執行檔

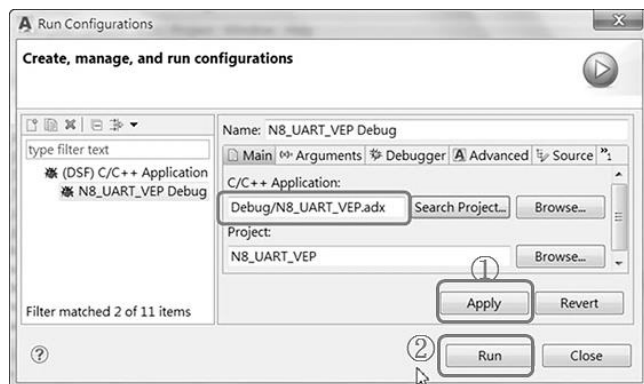


圖 5.3.4-12 完成執行設定檔

Step 10. 程式執行後請在「Target Manager」視窗中點擊「Running Target」一項前方三角標誌後(圖 5.3.4-13)可見所有已載入的虛擬平台或實體平台，此時可以看見左方有一紅色禁止進入的標誌(圖 5.3.4-14)表示該平台正在執行，在左方有一三角標誌請使用滑鼠左鍵點擊開啟(圖 5.3.4-14 ①)，此時可見該模擬平台提供的模擬元件，使用滑鼠左鍵快速點擊「Uart」(圖 5.3.4-14 ②)兩下，開啟 Uart 終端機模擬視窗(圖 5.3.4-15)。

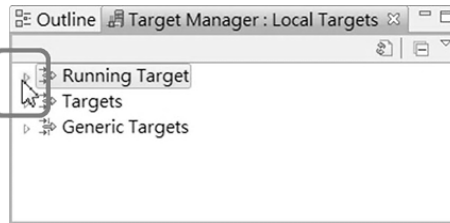


圖 5.3.4-13 開啟載入目標平台管理器裡正在執行的平台

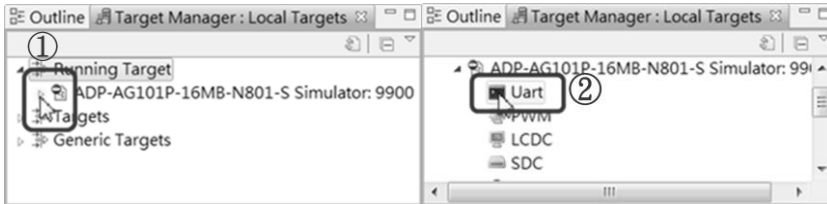


圖 5.3.4-14 開啟 VEP 選擇模擬元件 Uart

Step 11. 在 Step 10. 中開啟 Uart 模擬元件後，可以看到 Uart 的模擬終端機被開啟(圖 5.3.4-15)，且在其中顯示傳回的字串與變數「i」的值。

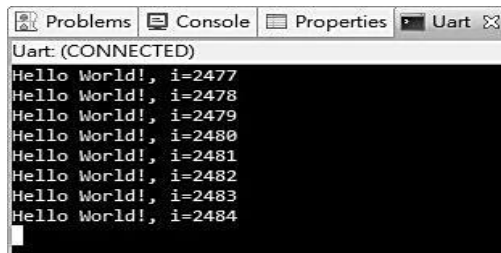


圖 5.3.4-15 Uart 模擬終端機

● N8_UART_XC5 實作

本實作改編自上一個實作，專案內容沒有作任何更動，只是重新設定工作平台，將程式碼透過 AICE 下載到 XC5 平台執行，以下是執行步驟：

Step 1. 複製 N8_UART_VEP 專案並重新改專案名稱，圖 5.3.4-16 標示執行步驟，首先將游標移至 N8_UART_VEP 專案名稱並以滑鼠左鍵點擊(圖 5.3.4-16 ①)之後按滑鼠右鍵出現下拉式選單以滑鼠左鍵點擊「Copy」(圖 5.3.4-16 ②)，再將滑鼠移至瀏覽視窗空白處按滑鼠右鍵出現下拉式選單，以滑鼠左鍵點擊「Paste」(圖 5.3.4-16 ③)完成專案複製。

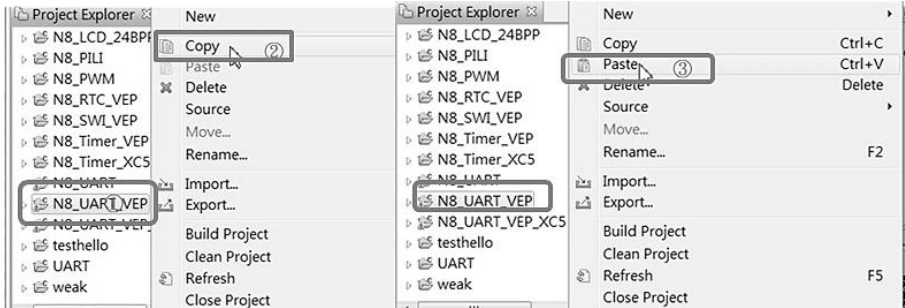


圖 5.3.4-16 拷貝專案並改專案名稱

Step 2. Step 1. 完成專案複製後出現圖 5.3.4-17 左方，將游標移至專案名稱「Project name」欄位處(①)將其改名為 N8_UART_XC5(②)，當然也可不改名略過此步驟。重新改專案名稱後點擊「OK」完成。

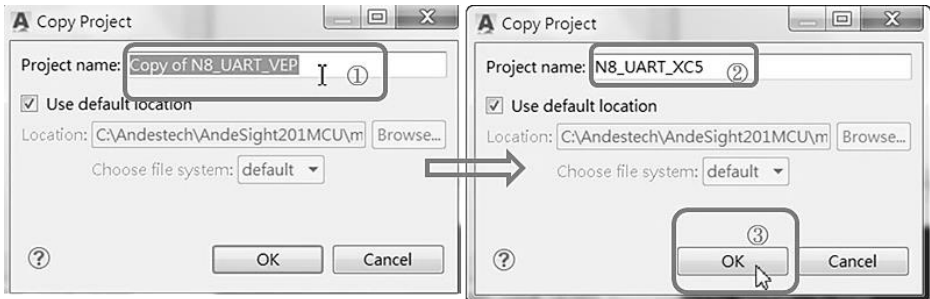


圖 5.3.4-17 改變專案名稱

»» 微處理器應用與實作：C 語言與 Andes MCU 系列

Step 3. 清除專案內已編譯連結的檔案，步驟如圖 5.3.4-18。因為原來的專案已編譯連結過，因此必須清除之，否則執行時會出現上一 Step 8. 的情形必須再一次選擇執行檔名專案。

Step 4. 重新設定專案執行平台。原來的專案設定執行的環境是模擬平台，但是此一專案要透過 AICE 下載到 XC5 平台執行，因此須重新設定平台及連結方式。圖 5.3.4-19 的執行步驟開啟設定平台及連結工具設定，圖 5.3.4-20 中第 ② 步驟事實上是依實際的工作平台選取，在此專案中無須更動，但是連結方式由「Simulator」改成「AICE」(圖 5.3.4-20 ③)。

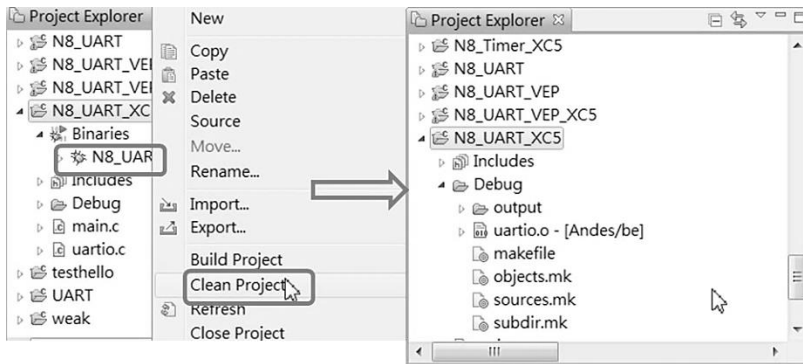


圖 5.3.4-18 清除專案編譯連結檔案

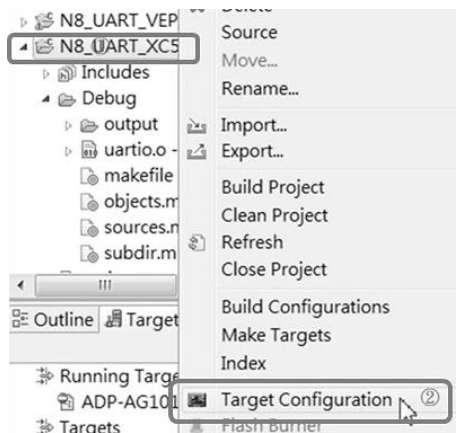


圖 5.3.4-19 選擇專案工作平台

Step 5. 更改最佳化。在 N8_UART_VEP 專案中編譯連結程式時，採用最佳化設定 (-Os)，編譯後可得最小的執行程式碼，但建議使用者在完成專案開發前最好不要進行最佳化，因為有些時最佳化的程式碼已不是原來的本意造成除錯時的困擾。圖 5.3.4-21 顯示開啟編譯環境設定步驟，圖 5.3.4-22 的執行步驟開啟最佳化參數設定視窗。圖 5.3.4-23 中顯示第 ② 步驟倒三角標誌可以打開下拉式選單，請選擇「-O0」取消最佳化的程序。

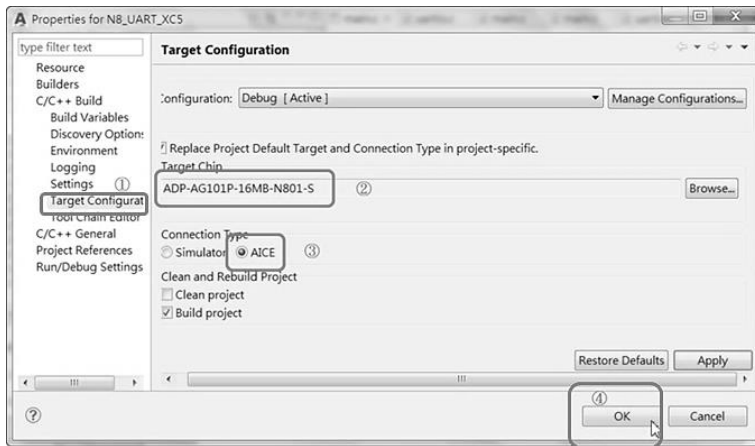


圖 5.3.4-20 設定工作平台及連結方式

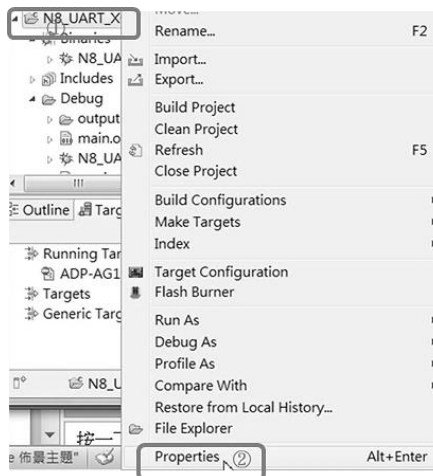


圖 5.3.4-21 開啟編譯環境設定

»» 微處理器應用與實作：C 語言與 Andes MCU 系列

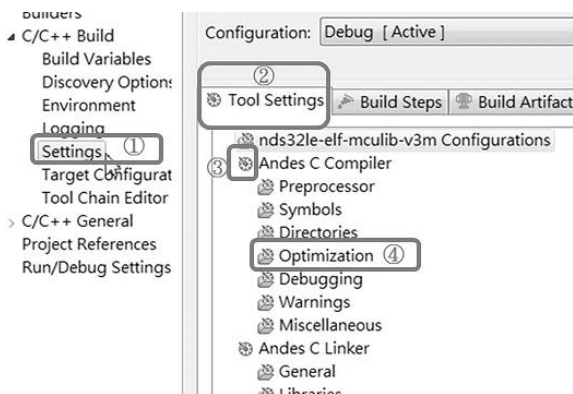


圖 5.3.4-22 編譯環境設定

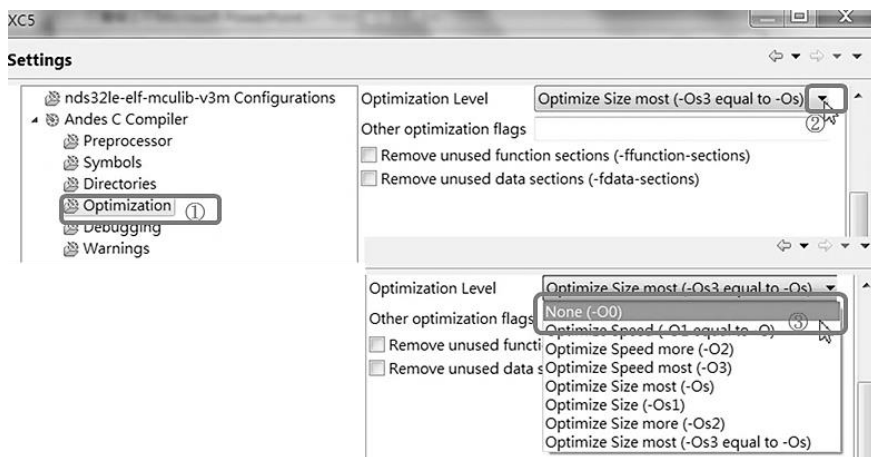


圖 5.3.4-23 最佳化參數設定

Step 6. 編譯連結的步驟請參考 N8_UART_VEP 專案 Step 7。專案建置完成後緊接下載程式碼到 XC5 平台執行，並透過終端機程式查看傳回終端機的資料是否與 N8_UART_VEP 專案執行結果一致。圖 5.3.4-24 中先點選執行的專案名稱（圖 5.3.4-24 ①）並將游標移至上方（②）綠圈白色三角符號的執行按鈕，此時可以看到出現專案執行檔名，可以確認要執行的檔案無誤，按下按鈕後，「Target Manager」視窗「Running Target」多一個正在執行的平台（圖 5.3.4-24 ③），且平台名稱後面顯示透過 AICE 經由 port 9903 與開

發主機連線。圖 5.3.4-25 是 XC5 平台執行 N8_UART_XC5 專案時透過 RS232 傳回開發主機終端機程式的畫面，這個畫面與 N8_UART_VEP 專案執行的畫面一致，只是使用不同的終端機程式而已。

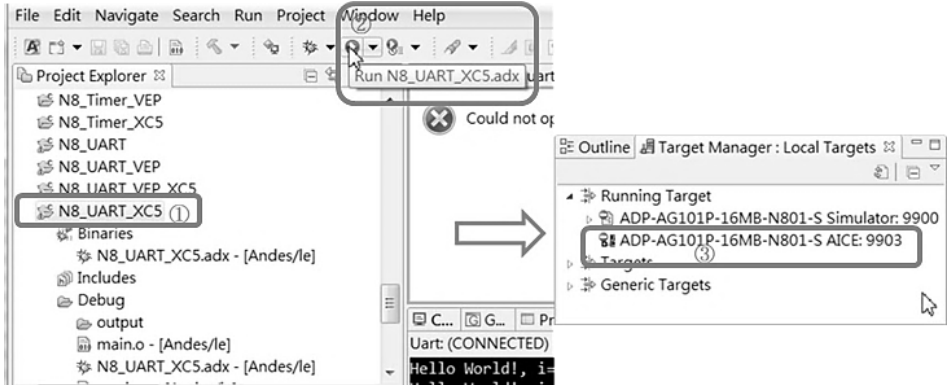


圖 5.3.4-24 下載程式碼到 XC5 平台執行

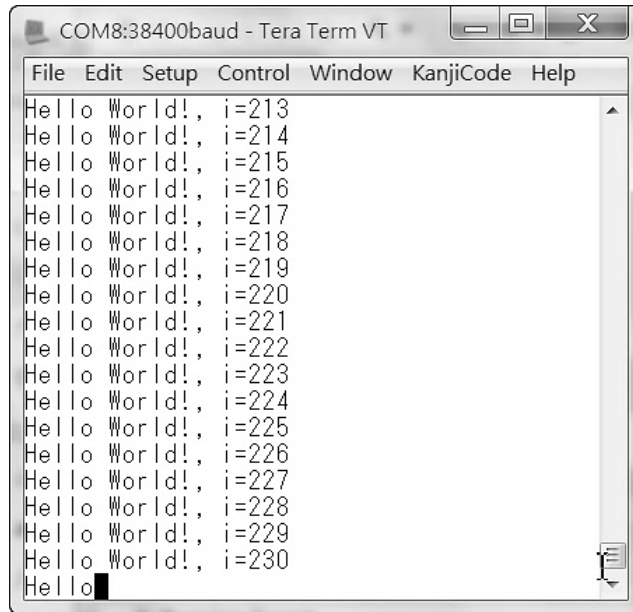


圖 5.3.4-25 XC5 傳送主機終端機程式的畫面