

序列埠通訊 (不使用中斷)

- 中斷使用起來雖方便，卻也危險
 - 中斷副程式必須朝精簡方向規劃，要不然一旦程式變多後會嚴重影響主流程運作
- 監看 SM4.5 和 SMB86 這兩個位址 (實現非中斷)
 - 以 S7-200 Programmable Controller System Manual 文件為基礎
 - TX: 當 SM4.5 = 1 (tx idle) ，執行 XMT 指令
 - RX: 執行 RCV 開始收資料並指定存放資料的起始位址，
 - 之後當 SMB86 = 0x20 時，表示收到正確的資料，第一個 byte 表示數量
 - 收到資料後，須將 SMB86 清成 0 才能讓後續動作正常 ???
 - 若出現錯誤的資訊，則？
 - 為何 RCV 指令只要執行一次，往後便能持續收資料 ???

序列埠的設定和指令

預設的 PPI 模式是下載程式時用的，須改成 Freeport 模式

Changing PPI Communications to Freeport Mode

SMB30 and SMB130 configure the communications ports, 0 and 1 respectively, for Freeport operation and provide selection of baud rate, parity, and number of data bits. Figure 6-8 describes the Freeport control byte. One stop bit is generated for all configurations.

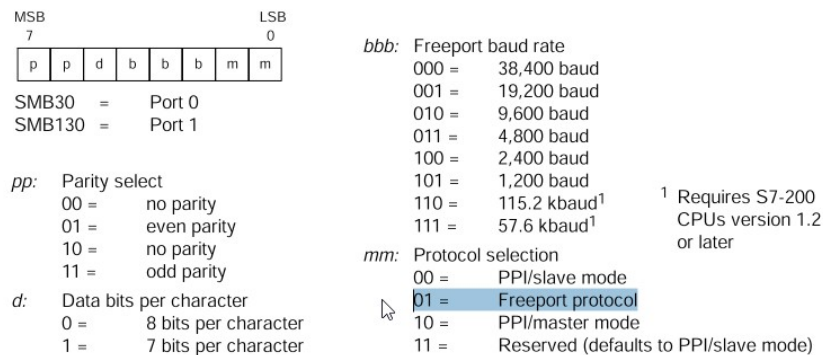


Figure 6-8 SM Control Byte for Freeport Mode (SMB30 or SMB130)

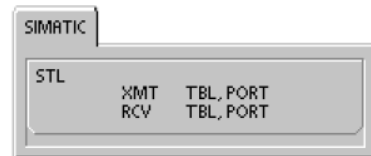
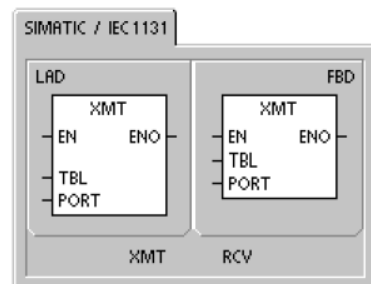
Transmit and Receive Instructions (Freeport)

The Transmit instruction (XMT) is used in Freeport mode to transmit data by means of the communications port(s).

The Receive instruction (RCV) initiates or terminates the receive message function. You must specify a start and an end condition for the Receive box to operate. Messages received through the specified port (PORT) are stored in the data buffer (TBL). The first entry in the data buffer specifies the number of bytes received.

Error conditions that set ENO = 0
data buffer 第一個 byte 表示 data bytes 數量，對 TX 和 RX 都適用

- 0006 (indirect address)
- 0009 (simultaneous Transmit/Receive on port 0)
- 000B (simultaneous Transmit/Receive on port 1)
- Receive parameter error sets SM86.6 or SM186.6
- S7-200 CPU is not in Freeport mode



從 PLC 傳資料給外面

Transmitting Data

The Transmit instruction lets you send a buffer of one or more characters, up to a maximum of 255.

Figure 6-9 shows the format of the Transmit buffer.

If an interrupt routine is attached to the transmit complete event, the S7-200 generates an interrupt (interrupt event 9 for port 0 and interrupt event 26 for port 1) after the last character of the buffer is sent.

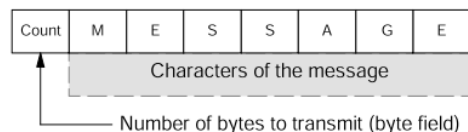


Figure 6-9 Format for the Transmit Buffer

You can make transmissions without using interrupts (for example, sending a message to a printer) by monitoring SM4.5 or SM4.6 to signal when transmission is complete.

You can use the Transmit instruction to generate a BREAK condition by setting the number of characters to zero and then executing the Transmit instruction. This generates a BREAK condition on the line for 16-bit times at the current baud rate. Transmitting a BREAK is handled in the same manner as transmitting any other message, in that a Transmit interrupt is generated when the BREAK is complete and SM4.5 or SM4.6 signals the current status of the Transmit operation.

SMB4: Queue Overflow

As described in Table D-5, SMB4 contains the interrupt queue overflow bits, a status indicator showing whether interrupts are enabled or disabled, and a transmitter-idle memory bit. The queue overflow bits indicate either that interrupts are happening at a rate greater than can be processed, or that interrupts were disabled with the global interrupt disable instruction.

Table D-5 Special Memory Byte SMB4 (SM4.0 to SM4.7)

SM Bits	Description (Read Only)
SM4.0 ¹	This bit is turned on when the communications interrupt queue has overflowed.
SM4.1 ¹	This bit is turned on when the input interrupt queue has overflowed.
SM4.2 ¹	This bit is turned on when the timed interrupt queue has overflowed.
SM4.3	This bit is turned on when a run-time programming problem is detected.
SM4.4	This bit reflects the global interrupt enable state. It is turned on when interrupts are enabled.
SM4.5	This bit is turned on when the transmitter is idle (Port 0).
SM4.6	This bit is turned on when the transmitter is idle (Port 1).
SM4.7	This bit is turned on when something is forced.

¹ Use status bits 4.0, 4.1, and 4.2 only in an interrupt routine. These status bits are reset when the queue is emptied, and control is returned to the main program.

西門子的文件對 SM4.5(用非中斷控制 TX) 的描述很少

從外面收資料回 PLC

Receiving Data

The Receive instruction lets you receive a buffer of one or more characters, up to a maximum of 255.

Figure 6-10 shows the format of the Receive buffer.

If an interrupt routine is attached to the receive message complete event, the S7-200 generates an interrupt (interrupt event 23 for port 0 and interrupt event 24 for port 1) after the last character of the buffer is received.

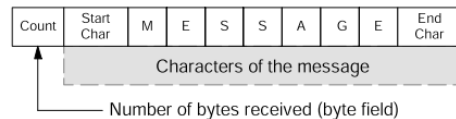
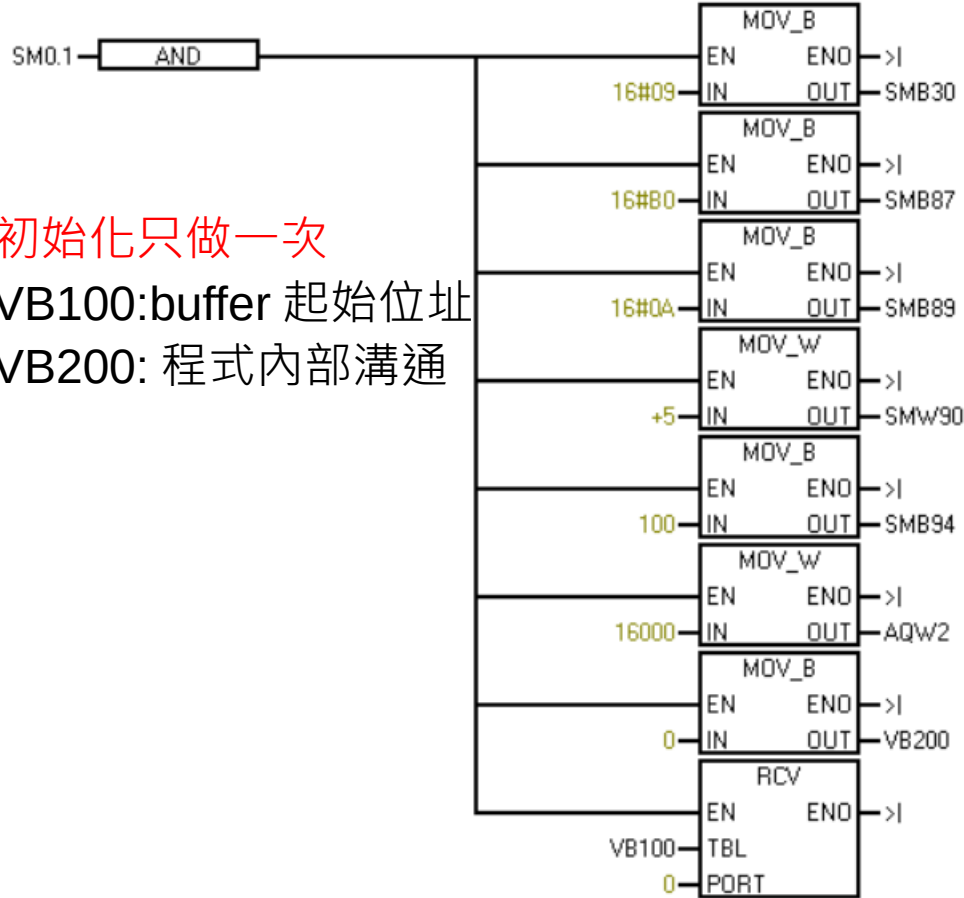


Figure 6-10 Format for the Receive Buffer

You can receive messages without using interrupts by monitoring SMB86 (port 0) or SMB186 (port 1). This byte is non-zero when the Receive instruction is inactive or has been terminated. It is zero when a receive is in progress.

Port 0	Port 1	Description										
SMB86	SMB186	Receive message status byte <table border="1" style="margin-left: 20px;"> <tr> <td style="text-align: right;">MSB 7</td> <td style="text-align: left;">LSB 0</td> </tr> <tr> <td style="text-align: center;">n</td> <td style="text-align: center;">r</td> </tr> <tr> <td style="text-align: center;">e</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">t</td> </tr> <tr> <td style="text-align: center;">c</td> <td style="text-align: center;">p</td> </tr> </table> <p>n: 1 = Receive message function terminated: user issued disable command.</p> <p>r: 1 = Receive message function terminated: error in input parameters or missing start or end condition.</p> <p>e: 1 = <u>End character received.</u></p> <p>t: 1 = Receive message function terminated: timer expired.</p> <p>c: 1 = Receive message function terminated: maximum character count achieved.</p> <p>p: 1 = Receive message function terminated: a parity error.</p>	MSB 7	LSB 0	n	r	e	0	0	t	c	p
MSB 7	LSB 0											
n	r											
e	0											
0	t											
c	p											
SMB87	SMB187	Receive message control byte <table border="1" style="margin-left: 20px;"> <tr> <td style="text-align: right;">MSB 7</td> <td style="text-align: left;">LSB 0</td> </tr> <tr> <td style="text-align: center;">en</td> <td style="text-align: center;">sc</td> </tr> <tr> <td style="text-align: center;">ec</td> <td style="text-align: center;">il</td> </tr> <tr> <td style="text-align: center;">c/m</td> <td style="text-align: center;">tmr</td> </tr> <tr> <td style="text-align: center;">bk</td> <td style="text-align: center;">0</td> </tr> </table> <p>en: 0 =Receive message function is disabled. <u>1 =Receive message function is enabled.</u> The enable/disable receive message bit is checked each time the RCV instruction is executed.</p> <p>sc: 0 =Ignore SMB88 or SMB188. 1 =Use the value of SMB88 or SMB188 to detect start of message.</p> <p>ec: 0 =Ignore SMB89 or SMB189. 1 =Use the value of SMB89 or SMB189 to detect end of message.</p> <p>il: 0 =Ignore SMW90 or SMW190. 1 =Use the value of SMW90 or SMW190 to detect an idle line condition.</p> <p>c/m: 0 =Timer is an inter-character timer. 1 =Timer is a message timer.</p> <p>tmr: 0 =Ignore SMW92 or SMW192. 1 =Terminate receive if the time period in SMW92 or SMW192 is exceeded.</p> <p>bk: 0 =Ignore break conditions. 1 =Use break condition as start of message detection.</p>	MSB 7	LSB 0	en	sc	ec	il	c/m	tmr	bk	0
MSB 7	LSB 0											
en	sc											
ec	il											
c/m	tmr											
bk	0											
X	SMB88	SMB188	Start of message character.									
O	SMB89	SMB189	End of message character.									
X	SMW90	SMW190	Idle line time period given in milliseconds. The first character received after idle line time has expired is the start of a new message.									
X	SMW92	SMW192	Inter-character/message timer time-out value given in milliseconds. If the time period is exceeded, the receive message function is terminated.									
O	SMB94	SMB194	Maximum number of characters to be received (1 to 255 bytes). This range must be set to the expected maximum buffer size, even if the character count message termination is not used.									

TX/RX Without Interrupt (by Polling)



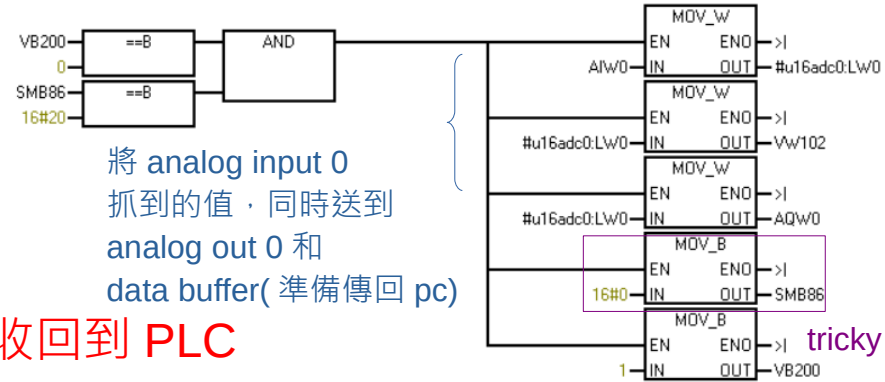
初始化只做一次
 VB100:buffer 起始位址
 VB200: 程式內部溝通

	Symbol	Var Type	Data Type	Comment
Lw0	u16adc0	TEMP	WORD	

暫存 analog input 0 轉換後的數位值

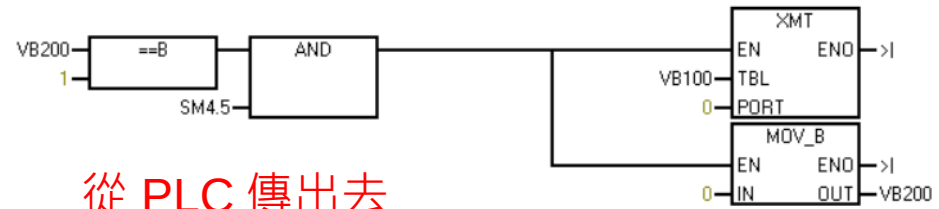
Network 3

save the value of analog input 0 in u16adc0[temporary?] and VM102[global]. Also, the same value is output to AQW0.



將 analog input 0 抓到的值 · 同時送到 analog out 0 和 data buffer(準備傳回 pc)

收回到 PLC



從 PLC 傳出去