MICROCHIP

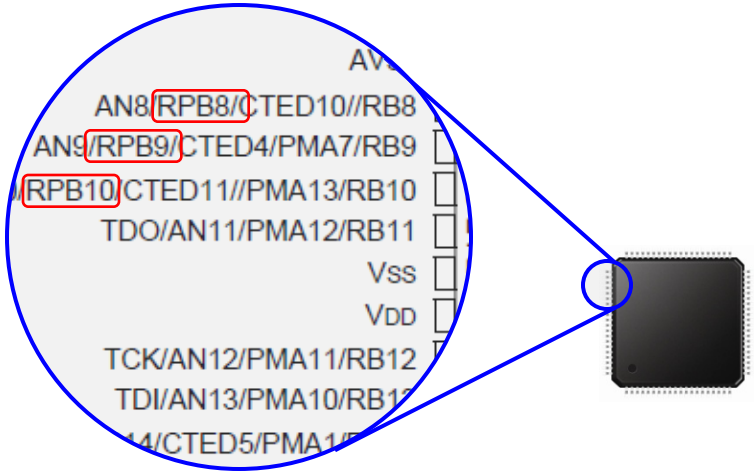
Regional Training Centers

Section 11
PPS and UART

MICROCHIP
Regional Training Centers

What's PPS

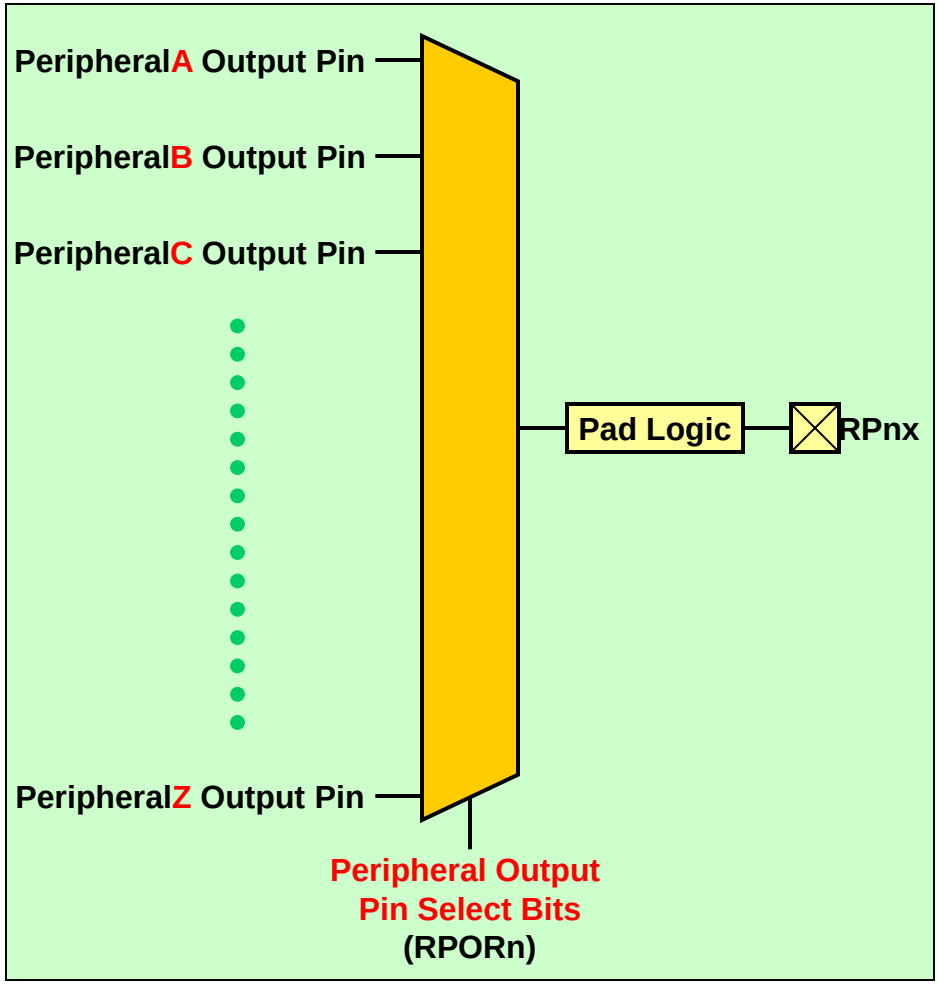
- PPS：Peripheral Pin Select,周邊接腳選擇
  一種可以重新改變周邊訊號接腳位置的功能。讓使用者可以自行安排(Remapping)周邊訊號接腳位置,達到最彈性化的接腳使用。
- PIC32MX450具有PPS功能。在接腳上有標示RP*nx*的, 都是可以Remapping的接腳。可已透過設定將UART,SPI,etc..的訊號接腳透過該Pin連接。
- PPS支援大部分的數位週邊,如:
  UART,SPI,OC,IC等。
- 但需要較複雜狀態處理的數位週邊及類比功能則不支援, 其接腳為固定不可變更的,如:
  I^2C,USB,PMP及類比數輸入等。



MICROCHIP
Regional Training Centers

PPS's Output

- 輸出多工器的輸入端連接到各個周邊的輸出訊號,輸出則接到特定接腳(RP*n*x)。
- 每個接腳(RP*n*x)都有專屬的多工器,可單獨指定其所要接的輸出訊號。
- PPS輸出的指定原則:
  指定接腳要使用哪個周邊的輸出。

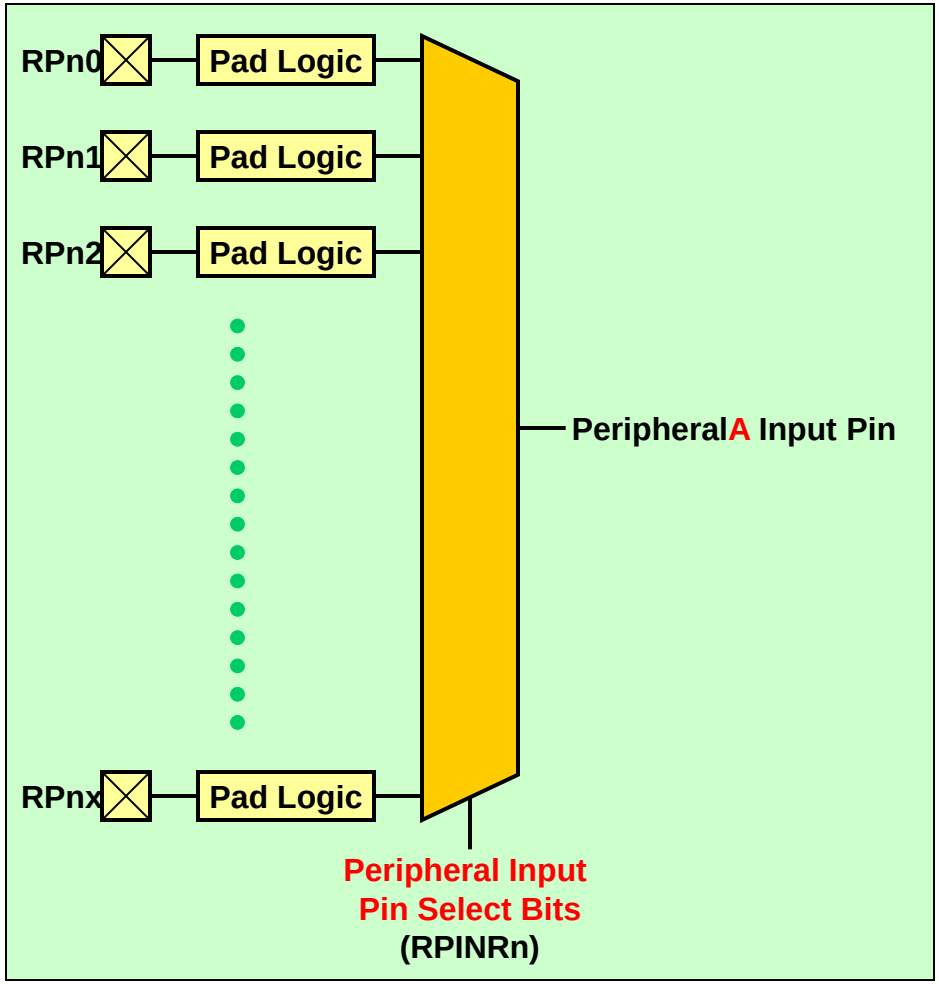


PPS基本輸出架構

MICROCHIP
Regional Training Centers

PPS's Input

- 輸入多工器的輸入端連接到各個輸入接腳(RP*n*x), 輸出則接到某個周邊的輸入接腳。
- 每個周邊都有專屬的多工器,可單獨指定其所要連接的接腳(RP*n*x) 。
- PPS輸入的指定原則:
  指定周邊的輸入要使用哪個接腳。



PPS基本輸入架構

MICROCHIP
Regional Training Centers

PPS Lock, Unlock

- PPS 設定前必須先進行解鎖,設定完後再上鎖。以避免非預期的動作更動PPS的設定。任何位元的非預期改變都會造成 MCU Reset。
- 解鎖動作必須將OSCCON的IOLOCK清除為0;

  OSCCONbits.IOLOCK = 0;

  鎖定動作必須必須將OSCCON的IOLOCK設為1。

  OSCCONbits.IOLOCK = 1;
- PPS 的鎖定模式,在Configuration Bit中設定

  IOL1WAY:IOLOCK One-Way Set Enable bit

  IOL1WAY = ON, PPS的Mapping只能被設定一次。

  IOL1WAY = OFF, PPS的Mapping可以重複的設定。

MICROCHIP
Regional Training Centers

XC32 PPS Function & Macro

- PPSOutput(*Group* , *RPPin* , *Function*);
  Group : PPS群組
  RPPin : PPS的接腳編號(RPn)
  Function : 周邊模組輸出訊號
  NULL , CxOUT , UxTX , UxRTS , SDOx , SCKxOUT , SSxOUT , OCx , CTMU
- PPSInput(*Group* , *Function* , *RPPin*);
  Group : PPS群組
  Function :周邊模組輸入訊號
  TxCK , INTx , ICx , OCFx , UxRX , UxCTS , SDIx , SCKxIN , SSxIN
  RPPin : PPS的接腳編號(RPn)
- PPSUnLock;
- PPSLock;

MICROCHIP
Regional Training Centers

PPS Group

- PIC32MX450的PPS有群組分配, 周邊功能與PPS接腳不能隨意分配, 必須遵照Datasheet的規劃。

TABLE 12-1: INPUT PIN SELECTION

Peripheral Pin	[pin name]R SFR	[pin name]R bits	[pin name]R Value to RPn Pin Selection
INT3	INT3R	INT3R<3:0>	0000 = RPD2 0001 = RPG8 0010 = RPF4 0011 = RPD10 0100 = RPF1 0101 = RPB9 0110 = RPB10 0111 = RPC14 1000 = RPB5 1001 = Reserved 1010 = RPC1(3) 1011 = RPD14(3) 1100 = RPG1(3) 1101 = RPA14(3) 1110 = Reserved 1111 = RPF2(1)
T2CK	T2CKR	T2CKR<3:0>	
IC3	IC3R	IC3R<3:0>	
U1RX	U1RXR	U1RXR<3:0>	
U2RX	U2RXR	U2RXR<3:0>	
$\overline{U5CTS}$	U5CTSR	U5CTSR<3:0>	
REFCLKI	REFCLKIR	REFCLKIR<3:0>	
INT4	INT4R	INT4R<3:0>	0000 = RPD3 0001 = RPG7 0010 = RPF5 0011 = RPD11 0100 = RPF0 0101 = RPB1 0110 = RPE5 0111 = RPC13 1000 = RPB3 1001 = Reserved 1010 = RPC4(3) 1011 = RPD15(3) 1100 = RPG0(3) 1101 = RPA15(3) 1110 = RPF2(1) 1111 = RPF7(2)
T5CK	T5CKR	T5CKR<3:0>	
IC4	IC4R	IC4R<3:0>	
U3RX	U3RXR	U3RXR<3:0>	
$\overline{U4CTS}$	U4CTSR	U4CTSR<3:0>	
SDI1	SDI1R	SDI1R<3:0>	
SDI2	SDI2R	SDI2R<3:0>	
INT2	INT2R	INT2R<3:0>	0000 = RPD9 0001 = RPG6

TABLE 12-2: OUTPUT PIN SELECTION

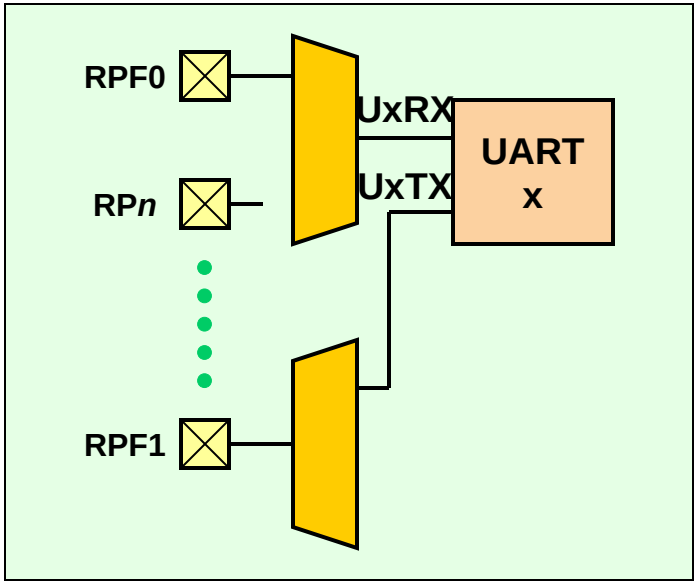
RPn Port Pin	RPnR SFR	RPnR bits	RPnR Value to Select
Group 1 Selections			
RPD2	RPD2R	RPD2R<3:0>	0000 = No Connec 0001 = U3TX 0010 = $\overline{U4RTS}$ 0011 = Reserved 0100 = Reserved 0101 = Reserved 0110 = SDO2 0111 = Reserved 1000 = Reserved 1001 = Reserved 1010 = Reserved 1011 = OC3 1100 = Reserved 1101 = C2OUT 1110 = Reserved 1111 = Reserved
RPG8	RPG8R	RPG8R<3:0>	
RPF4	RPF4R	RPF4R<3:0>	
RPD10	RPD10R	RPD10R<3:0>	
RPF1	RPF1R	RPF1R<3:0>	
RPB9	RPB9R	RPB9R<3:0>	
RPB10	RPB10R	RPB10R<3:0>	
RPC14	RPC14R	RPC14R<3:0>	
RPB5	RPB5R	RPB5R<3:0>	
RPC1(4)	RPC1R	RPC1R<3:0>	
RPD14(5)	RPD14R	RPD14R<3:0>	
RPG1(4)	RPG1R	RPG1R<3:0>	
RPA14(4)	RPA14R	RPA14R<3:0>	

PPS UART Mapping Example

- PPS設定UART2, Tx, Rx的Mapping:

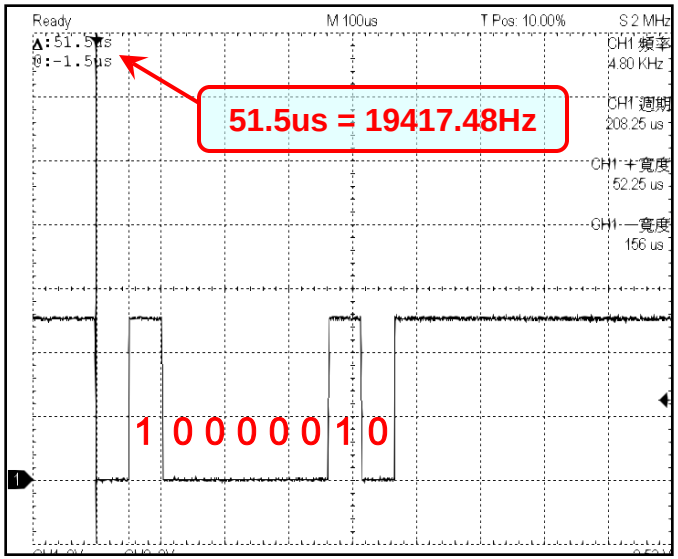
  Ex:

```
void main( void )
{
    ...
    ...
    ...
    ...
    ...
    PPSUnLock;
    PPSOutput(2, RPF0, U2TX);
    PPSInput(1, U2RX, RPF1);
    PPSLock;
    ...
}
```

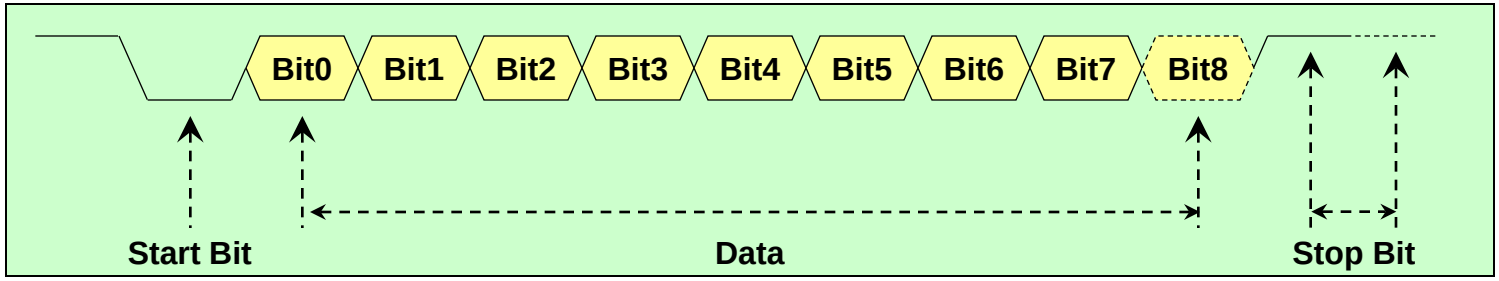


What's UART

- UART:Universal Asynchronous Receiver Transmitter,泛用非同步接收傳送模組。
  一個以串列方式進行資料交換與溝通的通訊模組。
- UART 傳輸的資料由LSB先傳。格式如下,包含一個起始位元,八或九位元的資料,一至二個停止位元。

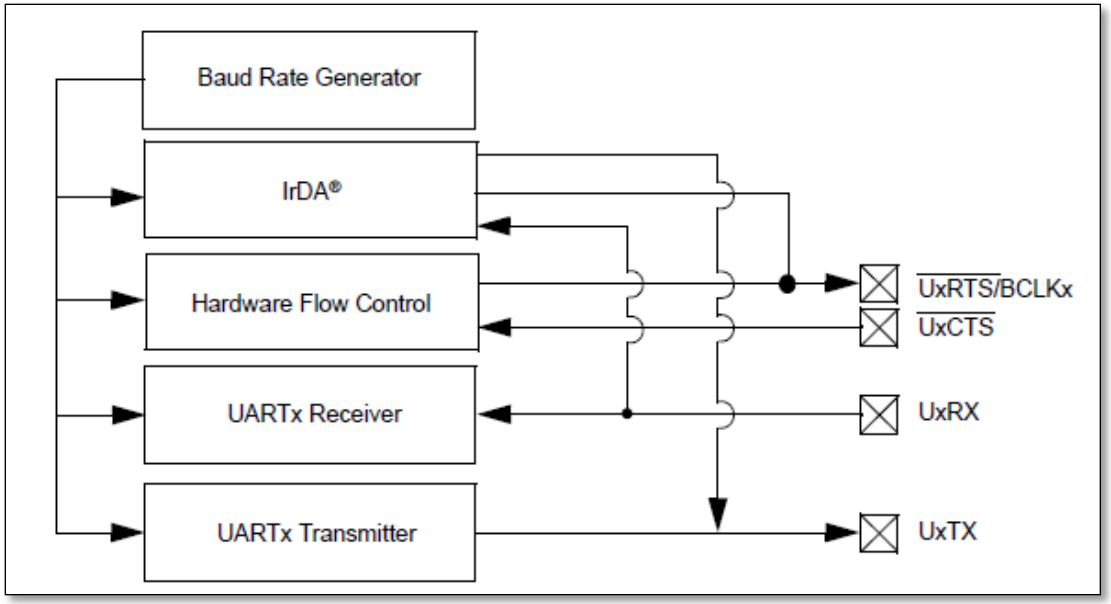


UART傳送範例'A'(0x41),19200 bps



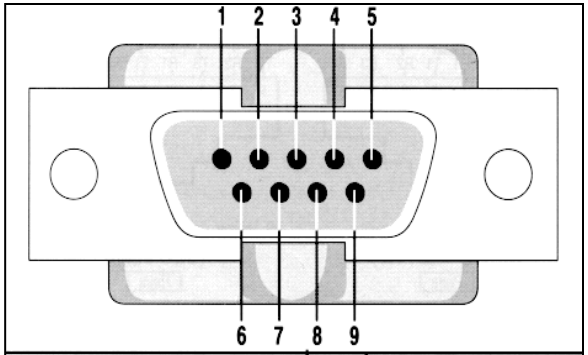
PIC32 UART Tran./Rece. Block

- PIC32的UART Module支援,8 or 9-Bits Data, 同位檢查及硬體流量控制(DTS,RTS)。
- 採全雙工, 非同步通訊。支援RS-232, RS-485, IrDA及LIN Bus。
- 八層深度的輸出入緩衝器(Buffer) 。
- 具有錯誤檢查機制(Parity Error, Frame Error, Overflow, etc..), Loopback及 Address Detect (RS-485), Auto Baud Detect(LIN bus)功能。



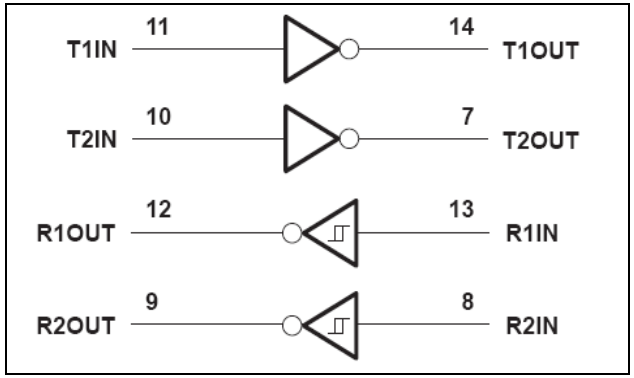
UART 與 RS232連接

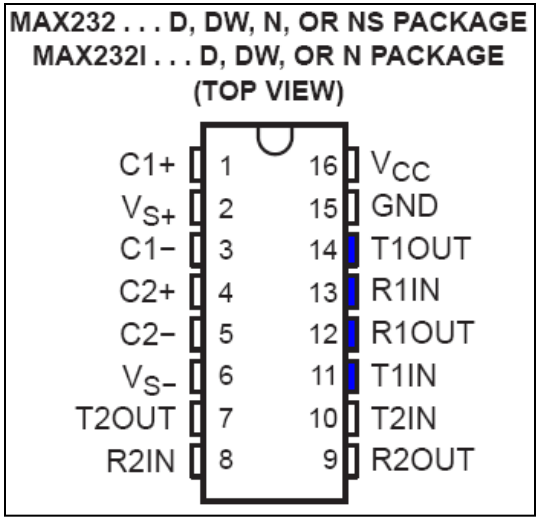
- RS232是一般PC常見的串列通訊埠。
- UART設成非同步傳輸, 透過位準與邏輯轉換後, 可與RS232裝置連接。
- RS232採用負邏輯, 與MCU正邏輯系統不同。正邏輯"1"的電壓準位為0.8 V_{DD} ~ V_{DD} ;邏輯"0"則為V_{SS} ~ 0.2V_{DD} 。負邏輯,邏輯"1"的電壓準位為-3~-12V;邏輯"0"則為+3 ~ +12V。因次必須使用UART與RS232連接時, 必須進行位準與邏輯轉換。



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

http://www.aggsoft.com/rs232-pinout-cable/serial-cable-connections.htm





MICROCHIP
Regional Training Centers

UART鮑率計算

- PIC32的鮑率(Baud Rate)的計算分為高低兩種速率,透過UxMODE中BRGH來設定。BRGH=0為低速;反之則為高速。
- 低速率時(BRGH=0), 計算方式如下:

$$UxBRG = \frac{SystemClock}{16 \times IdeaBaulRate} - 1 \qquad ActualBaudRate = \frac{SystemClock}{16 \times (UxBRG + 1)}$$

- 高速率時(BRGH=1),計算方式如下:

$$UxBRG = \frac{SystemClock}{4 \times IdeaBaulRate} - 1 \qquad ActualBaudRate = \frac{SystemClock}{4 \times (UxBRG + 1)}$$

MICROCHIP
Regional Training Centers

XC32 UART Function & Example

- **UART Initial Example**

```
UARTConfigure( UART2, UART_ENABLE_PINS_TX_RX_ONLY );
UARTSetFifoMode( UART2, UART_INTERRUPT_ON_TX_NOT_FULL |
                        UART_INTERRUPT_ON_RX_NOT_EMPTY );
UARTSetLineControl( UART2, UART_DATA_SIZE_8_BITS |
                           UART_PARITY_NONE | UART_STOP_BITS_1 );
UARTSetDataRate( UART2, PBFrequency, 19200 );
UARTEnable( UART2, UART_ENABLE | UART_PERIPHERAL |
                   UART_TX | UART_RX );
```

- **UART Interrupt Example**

```
INTSetVectorPriority( INT_UART_2_VECTOR, INT_PRIORITY_LEVEL_5 );
INTSetVectorSubPriority( INT_UART_2_VECTOR,
                                 INT_SUB_PRIORITY_LEVEL_0 );
INTEnable( INT_U2RX, INT_ENABLED );
```

MICROCHIP
Regional Training Centers

XC32 UART Function & Example

- **UART Receive (Polling) Example**

```
if( UARTReceivedDataIsAvailable(UART2 ) )
{
    Data = UARTGetDataByte( UART2 );
}
```

- **UART Transmit(Polling) Example**

```
if( UARTTransmitterIsReady( UART2 ) )
{
    UARTSendDataByte( UART2 , Data );
}
```

MICROCHIP
Regional Training Centers

XC32 UART Function & Example

- **UART Receive (Interrupt) Example**

```
void __ISR( _UART_2_VECTOR, ipl5 ) ISR_UART2(void)
{
        INTClearFlag( INT_U2RX );
        Data = UARTGetDataByte( UART2 );
}

INTSetVectorPriority( INT_UART_2_VECTOR, INT_PRIORITY_LEVEL_5 );
INTSetVectorSubPriority( INT_UART_2_VECTOR ,
                                INT_SUB_PRIORITY_LEVEL_0 );
INTEnable( INT_U2RX , INT_ENABLED );
```

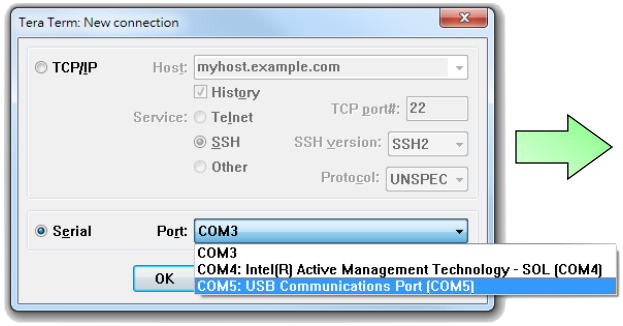
MICROCHIP
Regional Training Centers

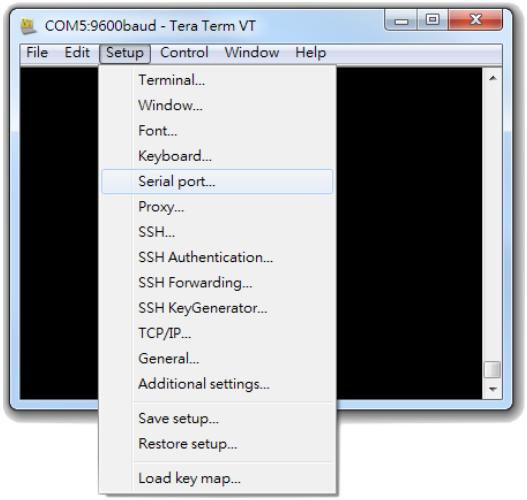
Lab11 UART Tx Polling Rx Interrupt

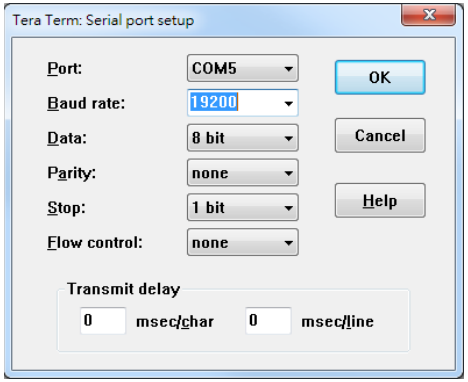
- 利用Lab8的程式,將UART的功能加入。透過UART將AD的數值傳送至PC。在超級終端機輸入的字元, 可以顯示在LCD Module上。
- 先初始化UART2, 為非同步傳輸, 規格為19200 , N , 8 , 1。建立UART2 接收的中斷服務常式 , 並開啟UART2的接收中斷, 設定優先權。優先權設定為”5” 。
- 設定PPS, 指定UART2的傳送接腳(U2TX)為RPF0, 接收(U2RX)為RPF1。
- 閱讀說明文件,了解UART Function與PPS Function的使用方法。
- 設定PC超級終端機軟體, 連接RS232 Cable。透過Bootloader將程式燒錄到MCU中, 再用肉眼觀察實驗板跟超級終端機軟體上的情況。

Tera Term

- 由於Windows 7不再提供超級終端機, 如果使用Windows必須必須先安裝替代軟體, 才可以監控COM Port的狀態。
- 請先安裝Tera Term, 然後開啟然後Tera Term, 指定COM Port, 設定為19200 , N , 8 , 1。







Lab11 UART Tx Polling Rx Interrupt

Step

- 超級終端機會看到輸出文字,每0.5秒更新一次。
- 在超級終端機上按任意鍵,會在LCD Module上面看到文字輸出。

MICROCHIP
Regional Training Centers

字串轉換

- 範例中的相關函式

```
unsigned char UARTString1[ 20 ];
sprintf( ( char * ) UARTString1 , "AD Value : %4d\r\n" , VR1Value );
 // 格式化輸出函數,功能與printf相同,唯一不同的是,
 // print輸出的目標是stdout(UART),sprintf則是輸出到一個字串陣列中。

void UART_PutRAMString( unsigned char * String )
{
        while( *String != 0x00 )
        {
                while( !UARTTransmitterIsReady( UART2 ) );
                UARTSendDataByte( UART2 , *String++ );
        }
}
// 將字串陣列的字元,逐一透過UART2傳送出去,直到字串結尾(0x00) 。
```

鮑率計算的誤差

- UART模組鮑率計算誤差?
  假設PB Clock為16MHz。預設鮑率115,200 bps。計算鮑率與誤差。
  低速率時(BRGH=0),鮑率與誤差計算:

$$\frac{16MHz}{16\times115200}-1=7.68\cong8 \qquad \frac{16MHz}{16\times(8+1)}\cong111111 \qquad \frac{111111-115200}{115200}\times100\%\cong-3.55\%$$

  高速率時(BRGH=1),鮑率與誤差計算:

$$\frac{16MHz}{4\times115200}-1=33.7\cong34 \qquad \frac{16MHz}{4\times(34+1)}\cong114286 \qquad \frac{114286-115200}{115200}\times100\%\cong-0.79\%$$

- UART誤差容忍度約為2%,在範例中,以低速率設定時,產生的誤差過高。此時就算UART的硬體設定無誤,也會因為過高的誤差導致UART動作不正常。