dsPIC30F Peripheral Module

dsPIC30F EUART Module

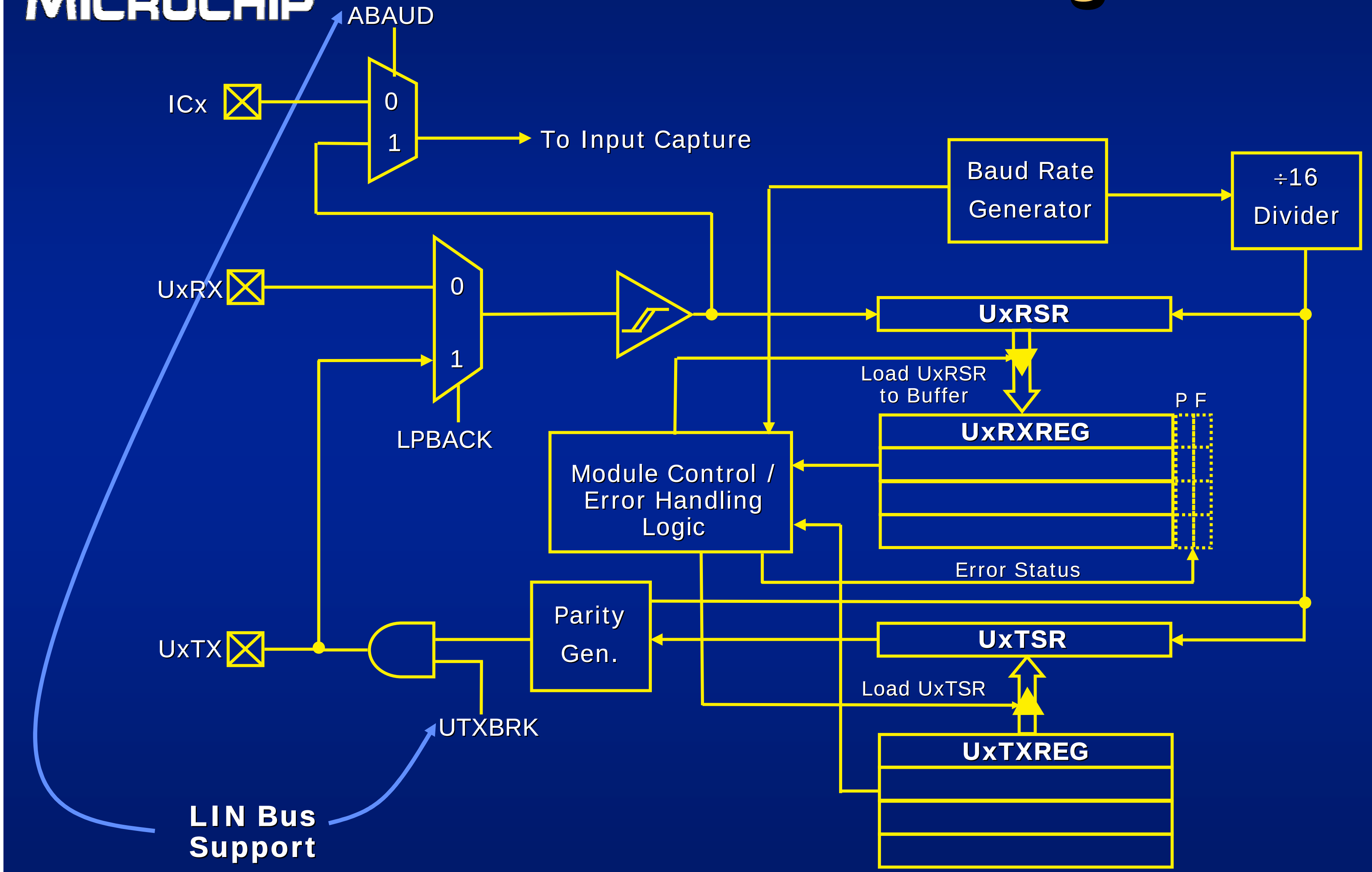
Session Agenda

- Module Overview
- UART Transmission
- UART Reception
- Additional Features

UART - Overview

- Serial transmission and reception
  - 8-bit (Odd, Even or No Parity) or 9-bit data
  - Full-duplex, asynchronous communication
  - Support for communication protocols such as RS-232, RS-422, RS-485 and LIN
  - 4-deep Transmit and Receive buffers
  - Transmit and Receive interrupts
  - Error detection
  - Support for receiver addressing
  - Loopback mode
  - Alternate TX/RX pins on some devices
  - Wake-up from SLEEP

UART - Block Diagram

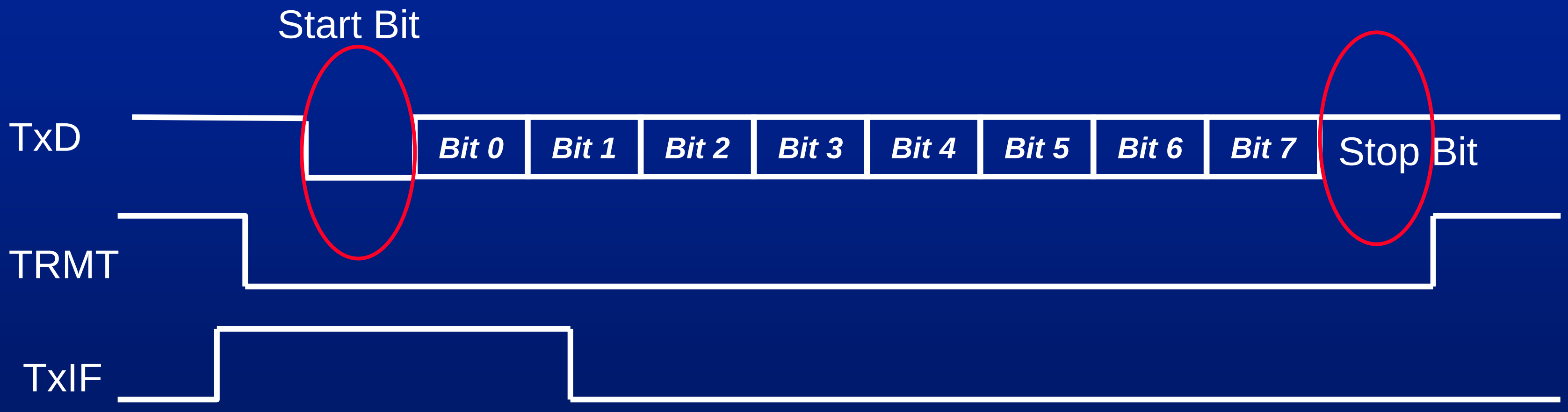


UART - Baud Rate Generator

- Dedicated 16-bit Baud Rate Generator
- Baud Rate controlled by UxBRG register (x = 1 or 2)
  - **Baud Rate = Fcy / (16 * (UxBRG + 1))**
    where Fcy = Instruction Cycle Frequency
- Both transmitting and receiving devices must use same Baud Rate
- Bits are transmitted and/or received at the rate defined by the Baud Rate

UART - Transmission

- UART module is enabled by setting the UARTEN bit in the UxMODE register
- Transmission starts only when:
  - The data to be transmitted is written to the buffer, AND UTXEN bit in the UxSTA register is set



UART – Transmission (contd.)

- UART module is enabled by setting the UARTEN bit in the UxMODE register

R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	U-0	U-0
UARTEN	-	USIDL	-	-	ALTIO	-	-
bit15	14	13	12	11	10	9	bit8

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
WAKE	LPBACK	ADAUD	-	-	PDSEL<1>	PDSEL<0>	STSEL
Bit 7	6	5	4	3	2	1	bit 0

- Transmission starts only when:
  - The data to be transmitted is written to the buffer, AND
  - The UTXEN bit in the UxSTA register is set

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R-0	R-1
UTXISEL	-	-	-	UTXBRK	**UTXEN**	UTXBF	TRMT
bit15	14	13	12	11	10	9	bit8

R/W-0	R/W-0	R/W-0	R-1	R-0	R-0	R/C-0	R-0
URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA
Bit 7	6	5	4	3	2	1	bit 0

UART - Transmission (contd.)

- The first bit transmitted is a START bit
  - Low level on UxTX pin for 1 bit time
- Next, the actual data bits are sent
  - LSB first , MSB later and parity bit last
  - Data format (8 or 9 bits) and parity type (even, odd or no parity) configured by PDSEL bits in the UxMODE register
  - No parity for 9-bit data

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
WAKE	LPBACK	ABAUD	-	-	**PDSEL1**	**PDSEL0**	STSEL
bit7	6	5	4	3	2	1	bit0

UART - Transmission (contd.)

- The last bit transmitted is a STOP bit
  - High level on UxTX pin for 1 or 2 bit times
  - Number of STOP bits configured by STSEL bit in the UxMODE register

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
WAKE	LPBACK	ABAUD	-	-	PDSEL1	PDSEL0	**STSEL**
bit7	6	5	4	3	2	1	bit0

- TRMT status bit in the UxSTA SFR
  - Bit is cleared if Transmit Shift Register (UxTSR) is busy or a transmission is pending

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R-0	R-1
UTXISEL	-	-	-	UTXBRK	UTXEN	UTXBF	**TRMT**
bit15	14	13	12	11	10	9	bit8

UART - Transmit Buffers

- 4-deep Transmit FIFO Buffer
  - ❖ Only the first character in the buffer is memory-mapped and thus user-accessible, UxTXREG
  - ❖ Characters in the buffer are shifted out of the buffer through UxTSR in FIFO
  - ❖ All 8 (or 9) data bits, are buffered
  - ❖ The UTXBF status bit in the UxSTA register indicates whether the buffer is full

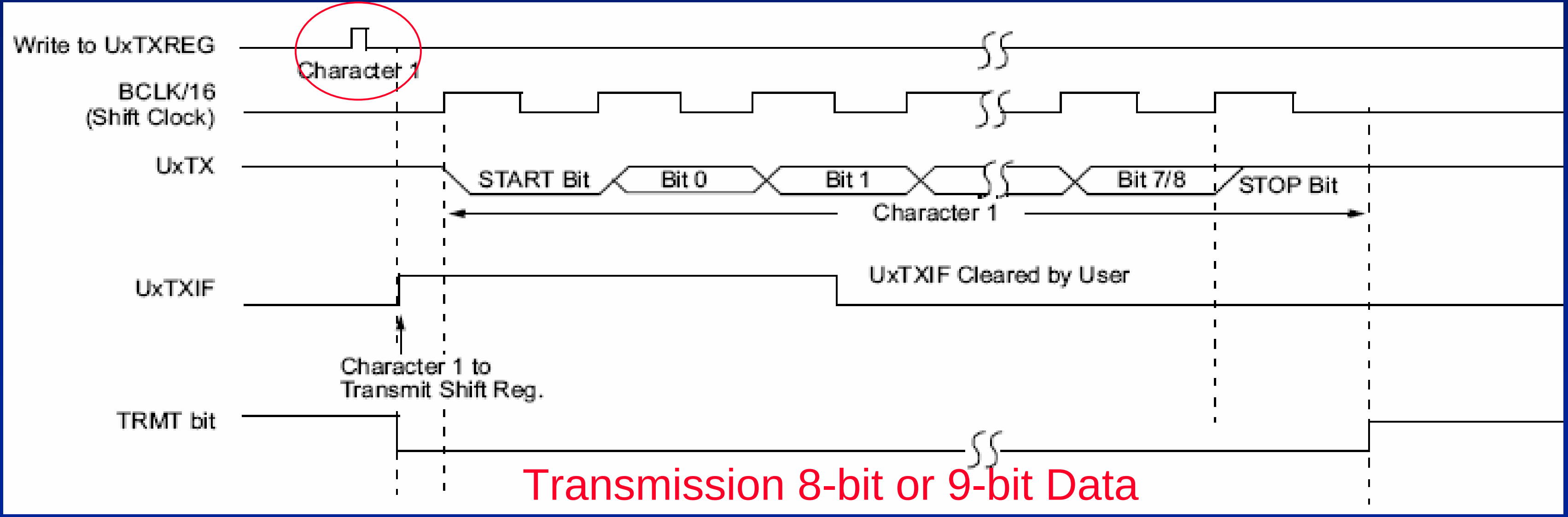
R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R-0	R-1
UTXISEL	-	-	-	UTXBRK	UTXEN	**UTXBF**	TRMT
bit15	14	13	12	11	10	9	bit8

UART - Transmit Interrupts

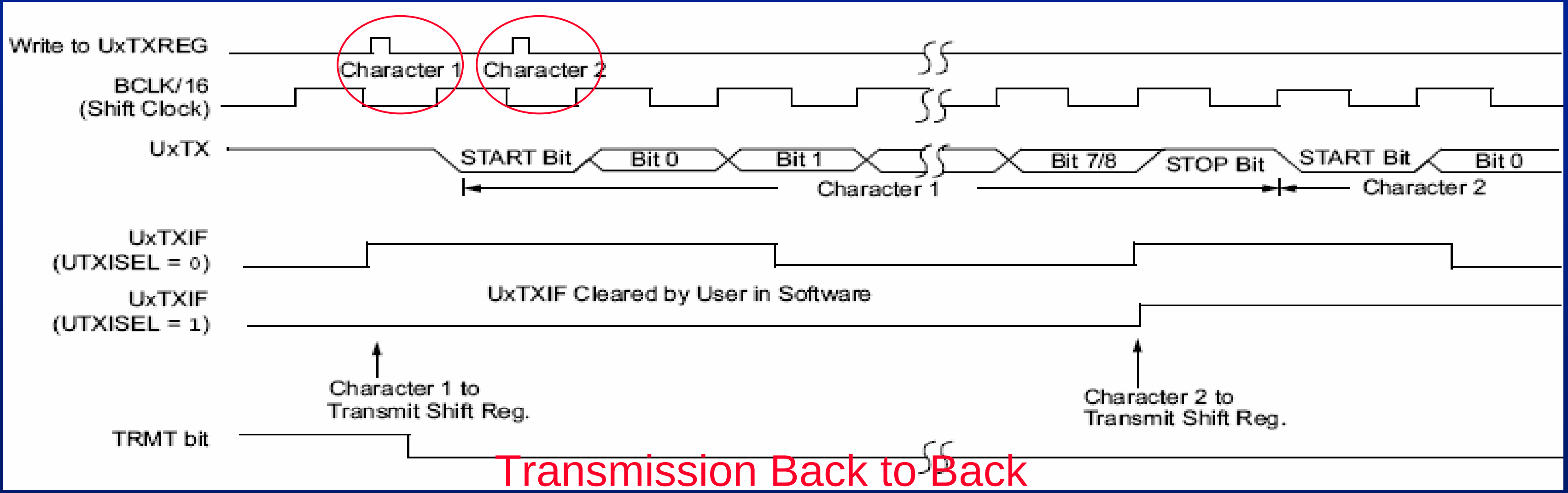
- Transmit Interrupt indicated by UxTXIF bit and enabled by UxTXIE bit
  - When UTXISEL bit in the UxSTA register = 1
    - Interrupt occurs when buffer becomes empty
    - Used for transmitting a block of 4 characters
  - When UTXISEL bit = 0
    - Interrupt occurs whenever a character is transferred to UxTSR
    - Used for transmitting a single character
    - In this mode, an interrupt is generated as soon as the UTXEN bit is set

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R-0	R-1
UTXISEL	-	-	-	UTXBRK	UTXEN	UTXBF	TRMT
bit15	14	13	12	11	10	9	bit8

TxD Interrupt Timing



Transmission 8-bit or 9-bit Data



Transmission Back to Back

UART Baud Rate Generator

- Dedicated 16-bit baud rate generator
  - *Baud Rate = Fcy / (16 * UxBRG+1)*

    or
  - *UxBRG = [FCY / (16 * Baud Rate)] – 1*

- Baud Rate Example
  - Fosc = 16MHz , Fcy=4MHz
  - Descried Baud Rate : 19200 bps

UxBRG = [4000000/(16*19200)] -1

= 13 -1 = 12

Baud Rate Error = 4000000/(16(12+1))=19230 bps

Error % = (19230-19200) / 19200 = 0.16%

UART - Reception

- UART Receiver becomes active when the module detects a START bit
- Data (starting with LSB) and parity bits are shifted through the Receive shift Register (UxRSR)
  - Data format and parity type (PDSEL bits in the UxMODE register) must be configured to match those of the transmitting device

UART - Reception (contd.)

- UxRSR stops shifting in bits when it detects a STOP bit
  - Number of STOP bits (STSEL bit in the UxMODE register) must be configured to match those of the transmitting device
- As long as UxRSR is shifting in bits, the module sets the RIDLE status bit in the UxSTA register
  - Bit remains clear at all other times

UART - Receive Buffers

- 4-deep Receive FIFO Buffer
  - All 8 (or 9) data bits, as well as Error flags, are buffered
  - Only the first character in the buffer is memory-mapped (UxRXREG)
  - The error flags in the UxSTA register reflect the error states of the first character in the buffer
  - URXDA status bit in the UxSTA register indicates if the buffer contains new data

UART - Receive Interrupts

- Receive Interrupt indicated by UxRXIF bit and enabled by UxRXIE bit
  - When URXISEL bits in the UxSTA register = 11
    - Interrupt occurs when buffer becomes full
    - Used for receiving a block of 4 characters
  - When URXISEL bits = 10
    - Interrupt when buffer has 3 characters
    - Used for receiving a block of 3 characters
  - When URXISEL bits = 01 or 00
    - Interrupt whenever a character is received
    - Used for receiving a single character

UART - Error Detection

- Parity Error
  - When received parity does not match the parity calculated by module from received data
  - Indicated by PERR bit in the UxSTA register set
- Framing Error
  - When a STOP bit is expected on UxRX pin but a low logic level is detected
  - Indicated by FERR bit in the UxSTA register set
- Receive Overrun Error
  - When the Receive Buffer is full and a 5th character is received
  - Indicated by OERR bit in the UxSTA register set

UART - Address Detection

- When the UART is operating in 9-bit mode (PDSEL = 11), and the ADDEN bit in the UxSTA register is set
  - The module will wait for an Address word, i.e., a 9-bit word with the 9th bit set
    - At this stage, the URXISEL bits in the UxSTA register must be set to 00 or 01
- On receiving the Address word, the user inspects the lower byte to verify an address match
- If an address match occurred, the user should clear the ADDEN bit, after which the module will wait for Data words (9-bit words with MSB clear)

UART - LIN Support

- Transmission of Break Characters
  - A Break character can be transmitted by setting the UTXBRK bit in the UxSTA register for at least 13 bit times

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R-0	R-1
UTXISEL	-	-	-	**UTXBRK**	UTXEN	UTXBF	TRMT
bit15	14	13	12	11	10	9	bit8

- Autobaud Detection
  - The UxRX pin is internally routed to an Input Capture pin (IC1 for UART1, IC2 for UART2)
  - Used for capturing both edges of START bit for determining baud rate

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
WAKE	LPBACK	**ABAUD**	-	-	PDSEL1	PDSEL0	STSEL
bit7	6	5	4	3	2	1	bit0

UART - Additional Features

- Alternate I/O
  - Some devices have an alternate pair of TX/RX pins
- Loopback Mode
  - UxTX pin internally connected to UxRX pin
- Wake-up from SLEEP
  - Device can be woken up from SLEEP by a START bit

R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	U-0	U-0
UARTEN	-	USIDL	-	-	**ALTIO**	-	-
bit15	14	13	12	11	10	9	bit8

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
WAKE	**LPBACK**	ABAUD	-	-	PDSEL1	PDSEL0	STSEL
bit7	6	5	4	3	2	1	bit0

Prepare UART Lab

- APP020 EVM Board setting request
  - DSW1 – SW1 & SW2 are ON position for the ICD2 programming & using the PGC, PGD for Debugging
  - DSW2 – SW1 & SW2 are ON position for UART1 (Lab1)

    SW3 & SW4 are ON position for UART2 (Lab2)
  - DSW3 – SW1 & SW2 are ON position for VR input
  - DSW4 – All OFF position
- Hyper-Terminal
  - Standard RS-232 communication port
  - Or, use the RS-232 to USB adapter
  - Set the Terminal : 9600 bps , Non parity, 8-bit data and 1 stop mode without any Handshake

Initialize the UART

```
CloseUART1( );                              /* Configure uart1 receive and transmit interrupt */
ConfigIntUART1 (UART_RX_INT_EN & UART_RX_INT_PR6 &
                UART_TX_INT_DIS & UART_TX_INT_PR2);
                                            /* Setup the Buad Rate Generator */
baudvalue = 95;                             //UxBRG = ( (FCY/Desired Baud Rate)/16) – 1
                                            //UxBRG = ( (7372800*2/9600)/16)-1 = 95

/* Configure UART1 module to transmit 8 bit data with one stopbit. Also Enable loopback mode */

U1MODEvalue = UART_EN & UART_IDLE_CON &
              UART_DIS_WAKE & UART_DIS_LOOPBACK &
              UART_DIS_ABAUD & UART_NO_PAR_8BIT &
              UART_1STOPBIT & UART_ALTRX_ALTTX;

U1STAvalue  =  UART_INT_TX_BUF_EMPTY &
               UART_TX_PIN_NORMAL &
               UART_TX_ENABLE & UART_INT_RX_CHAR &
               UART_ADR_DETECT_DIS &
               UART_RX_OVERRUN_CLEAR;

OpenUART1(U1MODEvalue, U1STAvalue, baudvalue);
```

Write data To UART1

- Use the WriteUART1 ( ) from Library

```
void WriteUART1(unsigned int data)
{
    if(U1MODEbits.PDSEL == 3)           // Check 9-bit data,no Parity
            U1TXREG = data;             // 9-bit transmission
    else
            U1TXREG = data & 0xFF;      // 8-bit transmission
}
```

- Use printf ( ) is more easily
  - Standard ANSI C syntax
  - Support the Float format
  - Extra code size

Read data from UART1

- Suggest to use the Interrupt Method
  - Why? It is real time and can run with background
  - Set U1RXIE bit in IEC0 register
  - Set the U1RXIP to a higher interrupt priority
- U1RXIF – UART1 Receiver Interrupt Flag
  - It should be cleared in Interrupt routine
  - This doesn’t like the PIC18, need to read the REREG to clear the RCIF flag
  - Refer to the URXISEL<1:0> control setting for receiver FIFO depth selection in the U1STA Register
- URXDA – Receiver Buffer Data Available bit
  - =1 : Receiver Data has unread buffer

Read data from UART1

- Using ReadUART1( )

```
unsigned int ReadUART1(void)
{
    if(U1MODEbits.PDSEL == 3)      // 9-bit data format?
        return (U1RXREG);
    else
        return (U1RXREG & 0xFF);
}
```

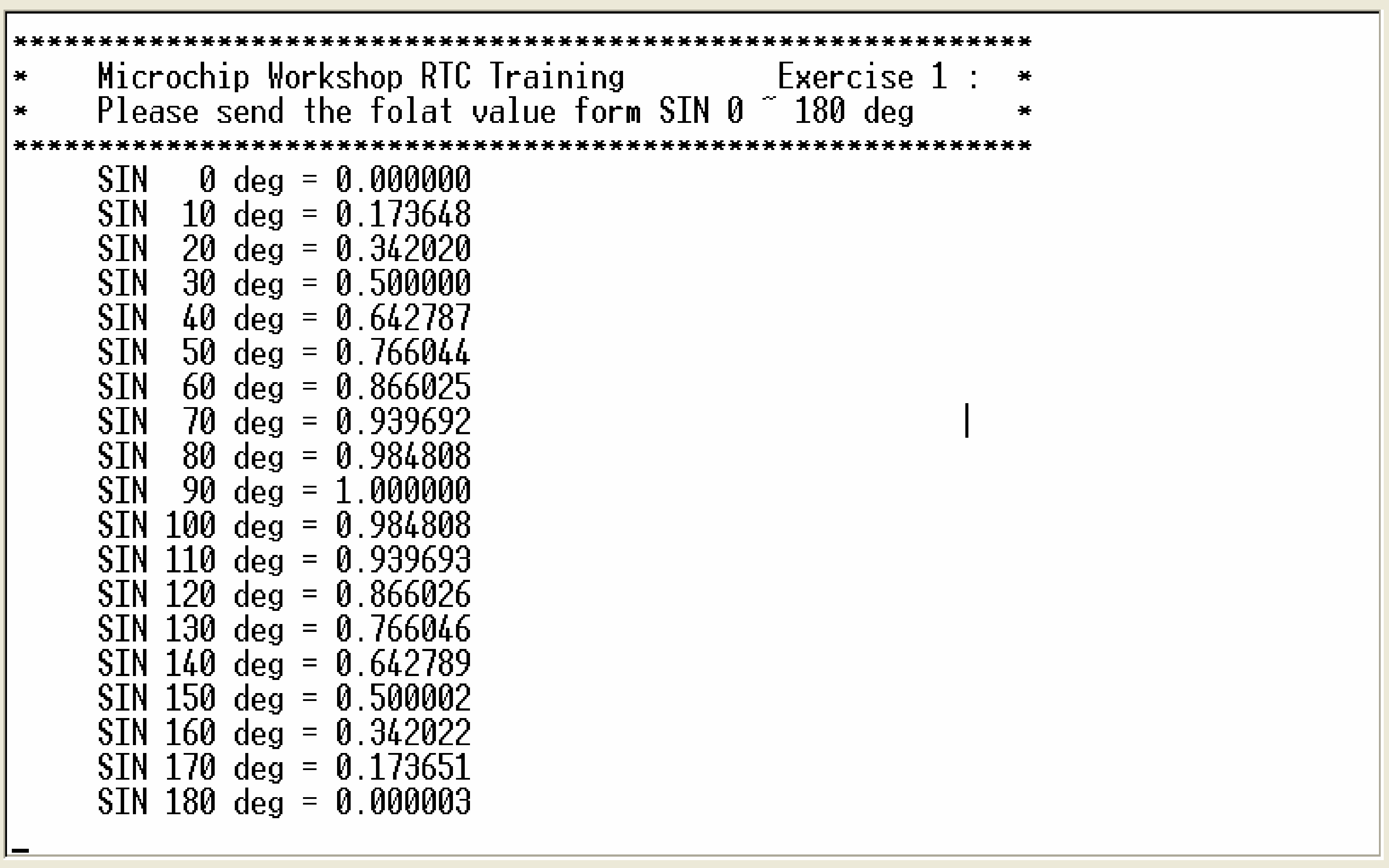
- Read receiver data from Interrupt

```
void _ISR _U1RXInterrupt(void)
{
    Rec_Buffer = ReadUART1( );   // Read data from Receiver FIFO
    Rec_Flag = 1;                // Set the Received Flag
    IFS0bits.U1RXIF = 0 ;        // Clear Interrupt Flag
}
```

UART Lab1

- Lab 1 can send a float data to the Hyper-Terminal through UART1
  - Set the communication protocol : 9600 pbs, no parity, 8 Data, 1 stop , non Handshake
  - Use the printf( )
- Lab1 can receive an ASCII code from the keyboard of Hyper-Terminal
  - Could be displayable ASCII code
  - Clear LCD when press the “Enter Key”
  - Display the string on the Line 2 of LCD

Hyper-Terminal Display



```
************************************************************
*    Microchip Workshop RTC Training         Exercise 1 :  *
*    Please send the folat value form SIN 0 ~ 180 deg      *
************************************************************
     SIN   0 deg = 0.000000
     SIN  10 deg = 0.173648
     SIN  20 deg = 0.342020
     SIN  30 deg = 0.500000
     SIN  40 deg = 0.642787
     SIN  50 deg = 0.766044
     SIN  60 deg = 0.866025
     SIN  70 deg = 0.939692
     SIN  80 deg = 0.984808
     SIN  90 deg = 1.000000
     SIN 100 deg = 0.984808
     SIN 110 deg = 0.939693
     SIN 120 deg = 0.866026
     SIN 130 deg = 0.766046
     SIN 140 deg = 0.642789
     SIN 150 deg = 0.500002
     SIN 160 deg = 0.342022
     SIN 170 deg = 0.173651
     SIN 180 deg = 0.000003
```

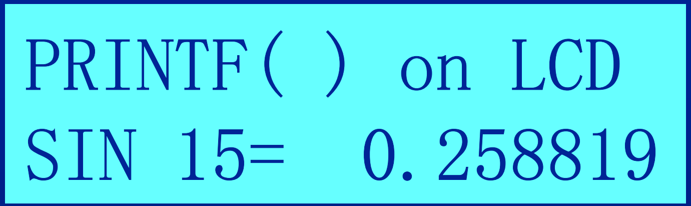
- Hyper-Terminal : 9600, N, 8, 1

UART Lab2

- Use the printf( ) to write the data to Hyper-Terminal through UART2
- Hint for Lab2 !!!
    - Function printf( ) default is using the UART1
    - Consider how to change the output to the UART2
    - Finally printf( ) will call the write.c
- Where can find the write.c
    - C:\Program Files\Microchip\MPLAB C30\src\pic30
- Modify the register in write.c for the UART2

Lab3 (options)

- Change the output to LCD using the printf( )
  - Modify the write.c direction to LCD
  - Using putcLCD (*(char*)buffer++);



```
PRINTF( ) on LCD
SIN 15=  0.258819
```

Lab 3 Display Result