Microchip Technology Inc.

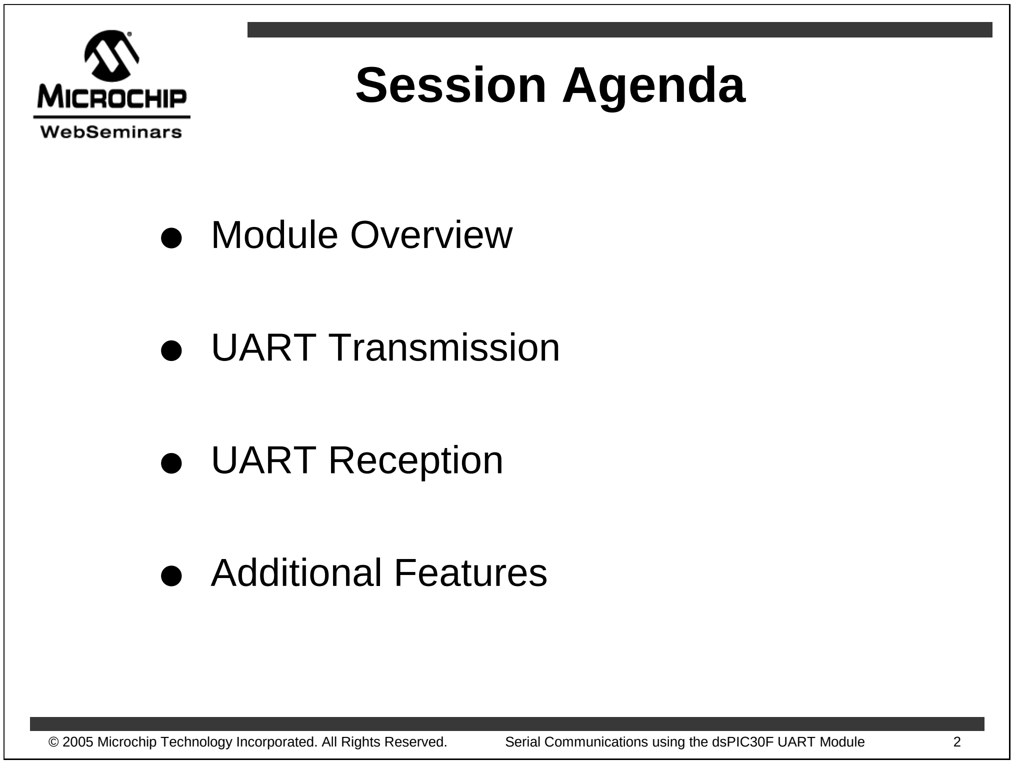
dsPIC™

Digital Signal Controller

Serial Communications using the dsPIC30F UART Module

Serial Communications using the dsPIC30F UART Module 1

Hello and welcome to the “Serial Communications using the dsPIC30F UART Module” web seminar. We’re glad that you’re here to learn more about Microchip’s family of Digital Signal Controllers.



Session Agenda

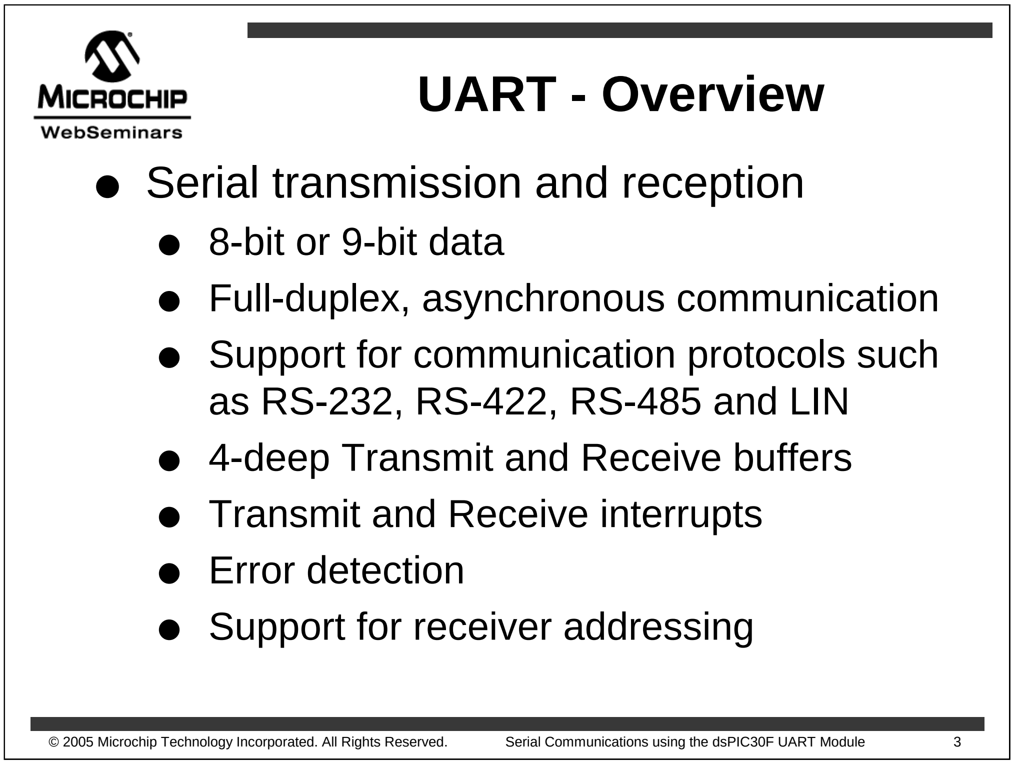
- Module Overview
- UART Transmission
- UART Reception
- Additional Features

My name is Ralph Fulchiero and I’m a Principle Applications Engineer with Microchip Technology.

In today’s session we’re going to learn about the Universal Asynchronous Receiver-Transmitter, commonly known as the UART module.

We will start by outlining some key features of the UART. We will gain an understanding of how the UART performs data transmission and data reception.

Lastly, we will study some advanced features of the UART module.



MICROCHIP WebSeminars

UART - Overview

- Serial transmission and reception
  - 8-bit 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

Serial Communications using the dsPIC30F UART Module 3

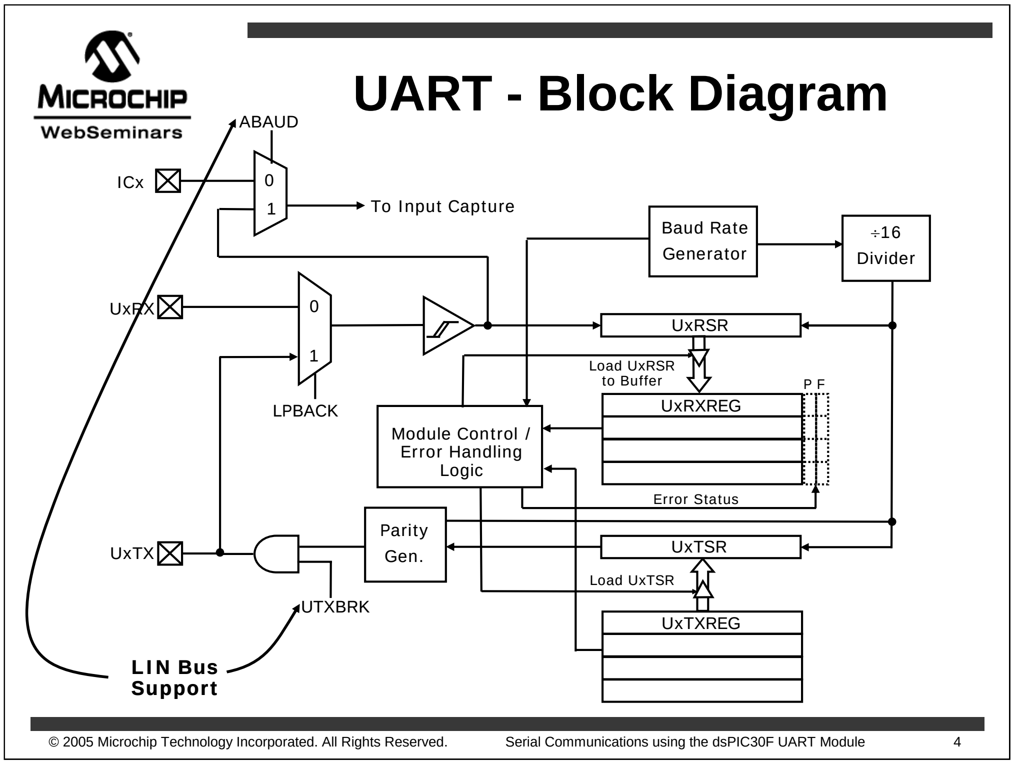
The UART module is used for transmission and reception of 8-bit or 9-bit data over a serial transmit line and a serial receive line. Many devices in the dsPIC30F family contain two UART modules and therefore have two pairs of transmit and receive pins.

UART communication is full-duplex, which means that a transmission and a reception can proceed simultaneously. It is also an example of Asynchronous communication, as each communicating entity is responsible for internally generating its own timing.

The UART module may be used with standard protocols such as RS-232, RS-422, RS-485, and Local Interconnect Network or LIN.

Besides basic transmission and reception capabilities, the UART has a number of useful features. For example, it includes independent buffering of up to 4 characters of transmitted data and up to 4 characters of received data, detection of communication errors, generation of interrupts, and the ability to address individual UART nodes.

Other features, which we will discuss in later slides, are Loopback, alternate communication pins, and Wake-up from SLEEP.



UART - Block Diagram

Serial Communications using the dsPIC30F UART Module 4

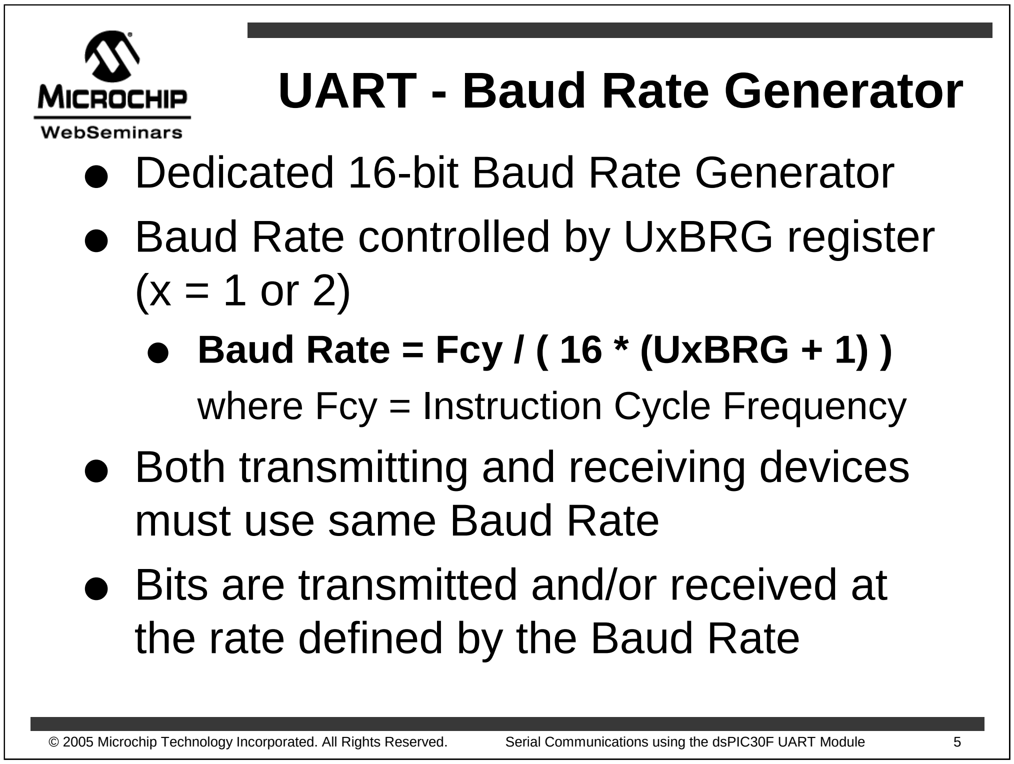
A simplified Block Diagram of the UART module is shown here. Note that the communications interface consists of only 2 pins: UART Transmit and UART Receive.

Observe also the 4-deep First In First Out buffering of data.

The Transmit Shift Register is used to shift out data through the UART Transmit pin, whereas the Receive Shift Register is used to shift in data through the UART Receive pin. Data in successive locations of the Transmit Buffer are transferred to the Transmit Shift Register, and Receive Shift Register contents are transferred to the Receive Buffer.

Only the first location of the Transmit Buffer is memory-mapped. It is user-accessible as the UART Transmit Register. Similarly, only the first received character of the Receive Buffer is user-accessible as the UART Receive Register.

There are some features provided mainly to support the LIN protocol. These features will be discussed later.



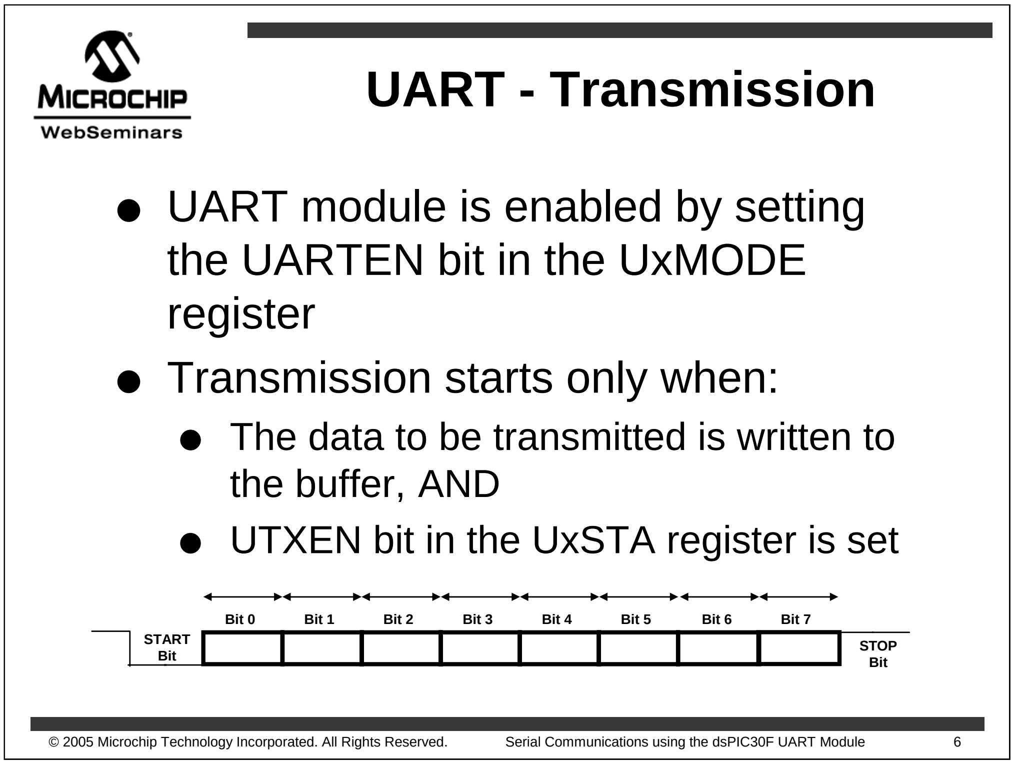
UART - Baud Rate Generator

- 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

Serial Communications using the dsPIC30F UART Module 5

The UART module contains a 16-bit Baud Rate Generator, called U1BRG or U2BRG, for UART1 and UART2 respectively. This register is used to provide timing synchronization of data transfer and all other UART events. The Baud Rate Generator is memory-mapped and therefore user-accessible.

Due to the communications being asynchronous, the user must configure the Baud Rate Generators of the transmitting device as well as the receiving device, such that both have the same Baud Rate. As is apparent from the equation, the maximum Baud Rate that can be attained is one-sixteenth of the instruction cycle frequency Fcy.



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

Bit 0
Bit 1
Bit 2
Bit 3
Bit 4
Bit 5
Bit 6
Bit 7
START Bit
STOP Bit

Serial Communications using the dsPIC30F UART Module 6

Before performing any transmissions or receptions, the UART module must be enabled by setting the UART Enable bit in the UART Mode Register.

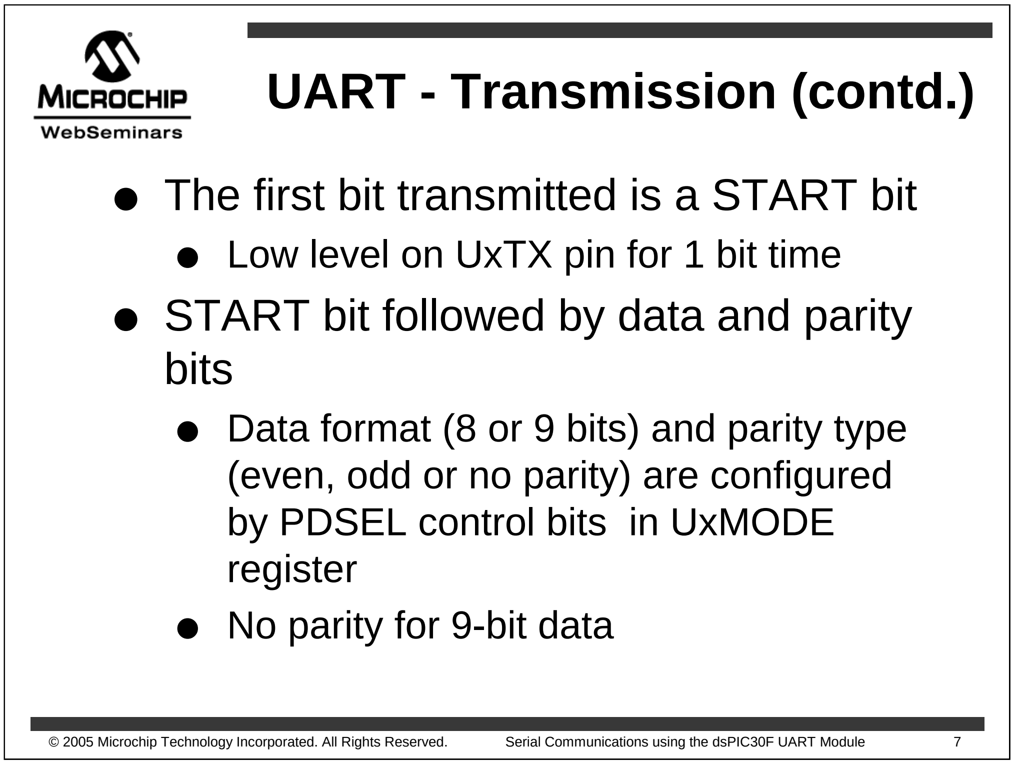
A simplified timing diagram of a typical UART transmission is illustrated here. Transmission of data can be initiated in one of 2 ways:

By first writing data to the UART Transmit Register and then setting the UART Transmit Enable bit in the UART Status Register,

or

By first setting the UART Transmit Enable bit and then writing to the UART Transmit Register register.

Clearing the UART Transmit Enable bit while a transmission is in progress, will abort any ongoing and pending transmissions, reset the transmitter, and bring the UART Transmit pin to a high impedance state.



MICROCHIP WebSeminars

UART - Transmission (contd.)

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

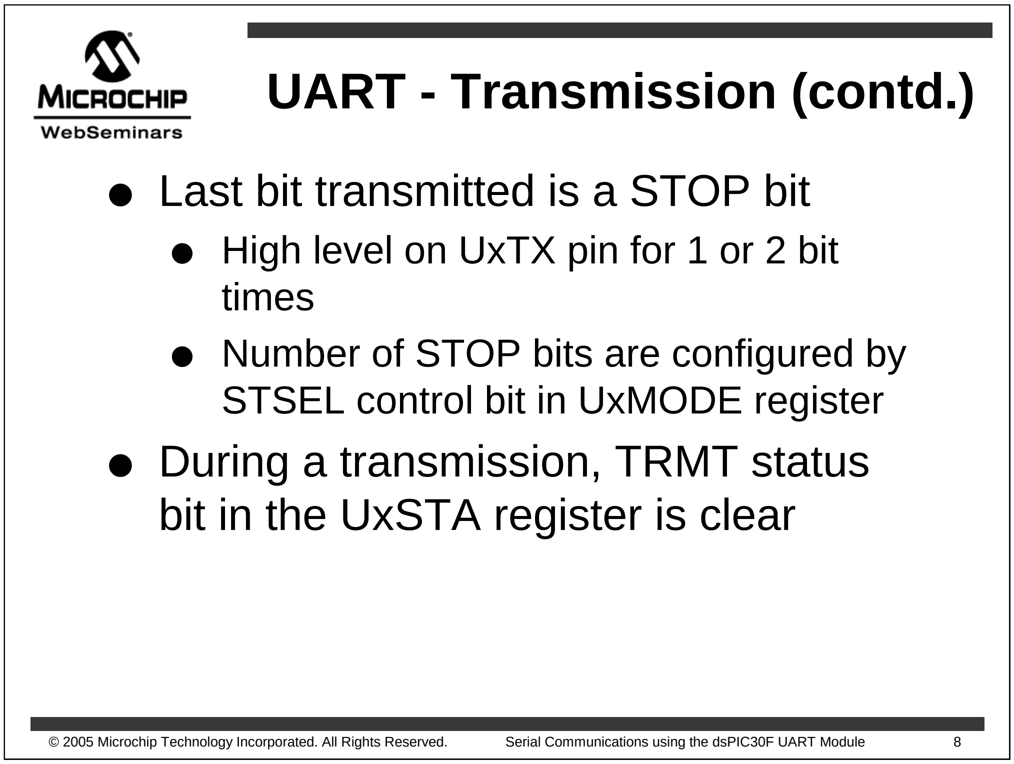
As soon as the data in the UART Transmit Register gets transferred to the Transmit Shift Register, the process of transmission begins with the transmitting device driving its Transmit line low for one bit time. This is known as a START bit, and indicates to the receiving device that the transmission of data bits is about to begin.

Following the START bit, the data bits are transmitted, with the Transmit line reflecting the value of each bit for a duration of one bit time. The bits are transmitted starting with the Least Significant Bit, or LSB, and ending with the Most Significant Bit, or MSB.

The UART module supports various data formatting options which are set in the UMODE register using the Parity and Data Select bits. The data length can be 8 or 9 bits. In the case of 9-bit data, a word write is required to the UART Transmit Register.

Further, the module can generate an even or odd parity in the case of 8-bit data, or disable parity generation for 8-bit data.

The parity bit denotes whether there is an even or odd number of ones in the transmitted character, and can be subsequently used by the receiving device to detect transmission errors.



UART - Transmission (contd.)

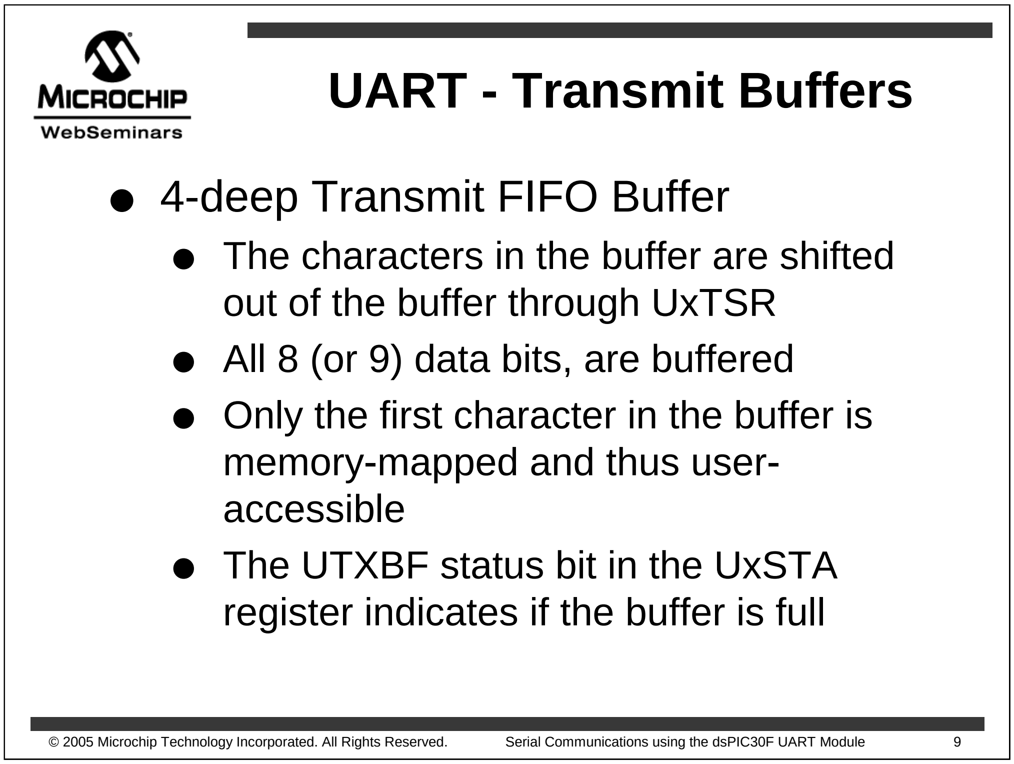
- Last bit transmitted is a STOP bit
  - High level on UxTX pin for 1 or 2 bit times
  - Number of STOP bits are configured by STSEL control bit in UxMODE register
- During a transmission, TRMT status bit in the UxSTA register is clear

© 2005 Microchip Technology Incorporated. All Rights Reserved. Serial Communications using the dsPIC30F UART Module 8

After all the 8 or 9 bits of data are transmitted, the end of the message is indicated when the transmitting device drives the Transmit line high for one or two bit times. These are known as STOP bits.

If the Stop Select bit in the UART Mode Register is set, then two STOP bits are generated, otherwise only one STOP bit is generated.

When the Transmit Shift Register Empty status bit in the UART Status Register is clear, it indicates that a transmission is either ongoing or pending. When the Transmit Shift Register as well as the Transmit Buffer are empty, then this bit is automatically set by the hardware.



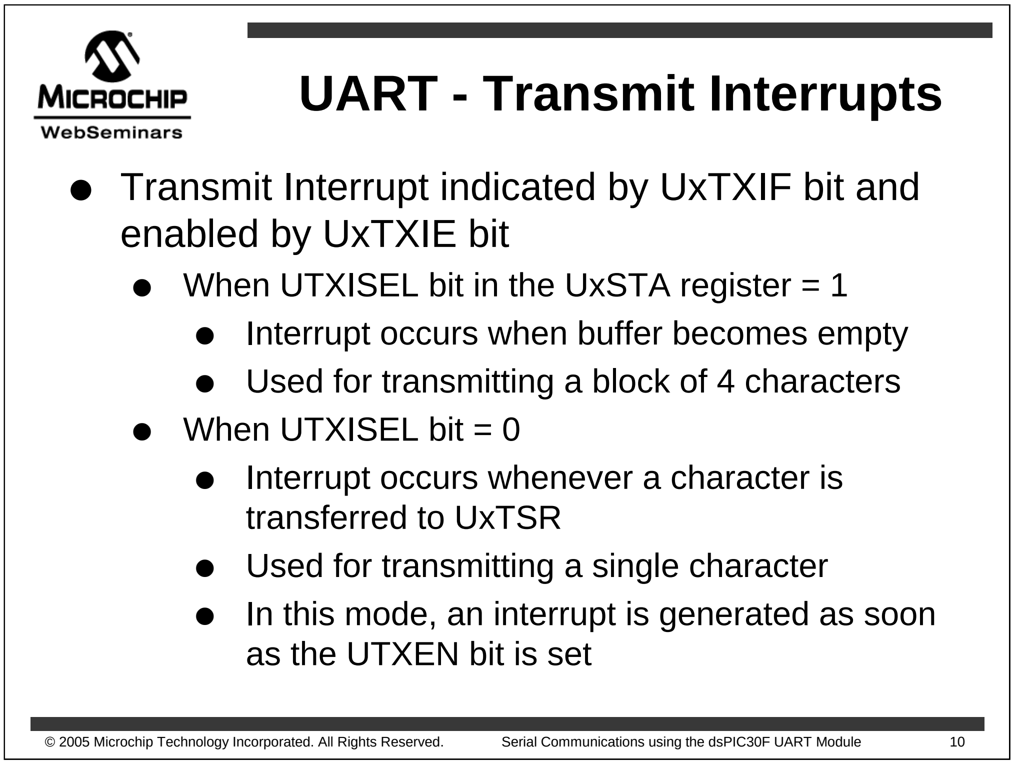
UART - Transmit Buffers

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

© 2005 Microchip Technology Incorporated. All Rights Reserved. Serial Communications using the dsPIC30F UART Module 9

As mentioned earlier, the Transmit Buffer is a 4-level deep FIFO buffer, of which only the first level is user-accessible. When a character is transferred from the Transmit Buffer to the Transmit Shift Register, the remaining characters are automatically moved up one level within the buffer, thereby ensuring that the first buffer location always contains the oldest character in the buffer and the characters are transmitted in the same sequence in which they were written by the user.

The UART Transmit Buffer Full status bit in the UART Status Register indicates whether the Transmit Buffer is full. The user may write to the Transmit Buffer only when this bit is clear, otherwise the write will be ignored by the hardware.



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

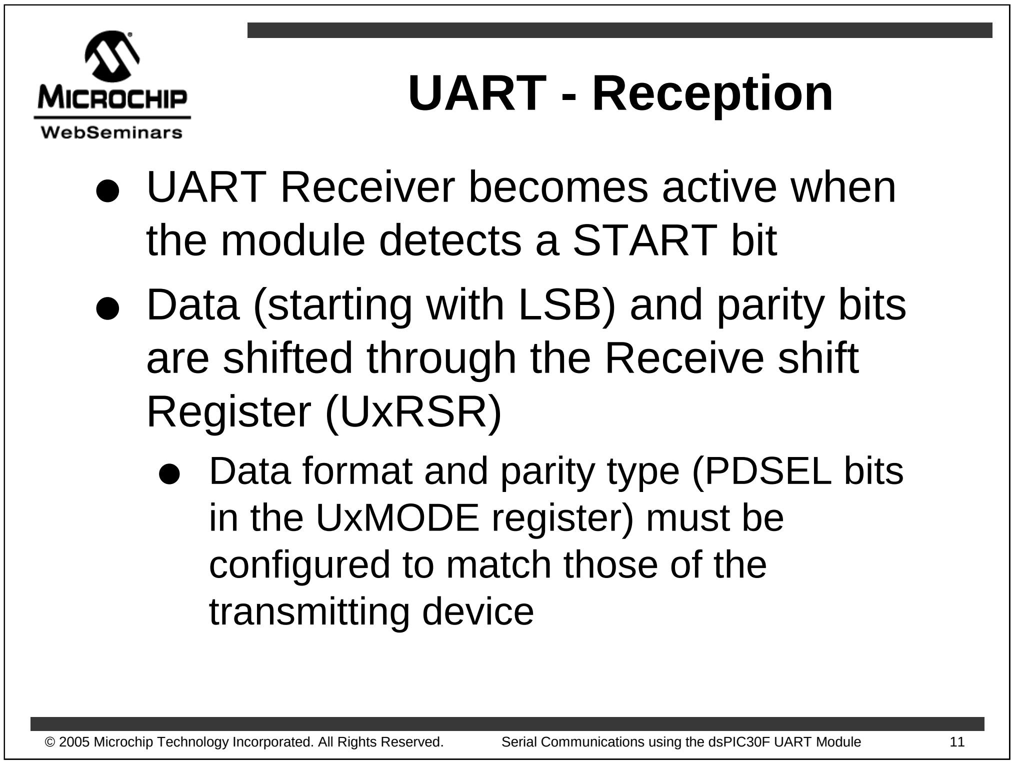
© 2005 Microchip Technology Incorporated. All Rights Reserved. Serial Communications using the dsPIC30F UART Module 10

The UART module can generate a UART Transmit Interrupt, which will be serviced if the corresponding interrupt enable bit is set and the interrupt priority is higher than the current CPU interrupt priority.

There are two Transmit Interrupt modes:

If the UART Transmit Interrupt Mode Select bit is set, then an interrupt is generated whenever a character transfer from the Transmit Buffer to the Transmit Shift Register causes the Transmit Buffer to become empty. This mode is useful for transmitting a block of 4 characters, especially in applications in which generating an interrupt for each character may not be desirable.

If the UART Transmit Interrupt Mode Select bit is cleared, then an interrupt is generated every time a character is transferred from the Transmit Buffer to the Transmit Shift Register. In other words, an interrupt is generated when at least 1 location is available in the Transmit Buffer. It may be inferred that this mode is useful when the application can tolerate the overhead of servicing an interrupt for each character to be transmitted. Note that in this mode, an interrupt is generated when the UART Transmit Enable bit is set for the first time.



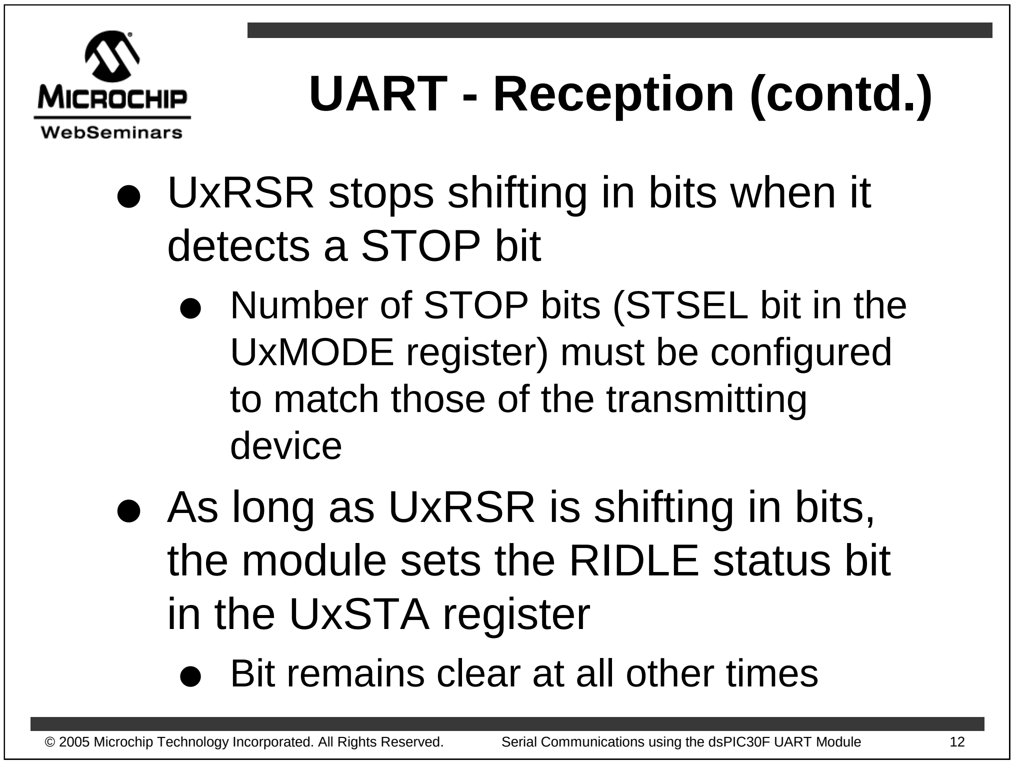
UART - Reception

- 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

Serial Communications using the dsPIC30F UART Module 11

When the receiving device detects a START bit in its UART Receive line, it starts shifting in data through the UART Receive Shift Register, starting with the LSB and ending with the MSB.

It is critical that the receiving device be configured for the same data format and parity as the transmitting device, in order to ensure correct reception.



UART - Reception (contd.)

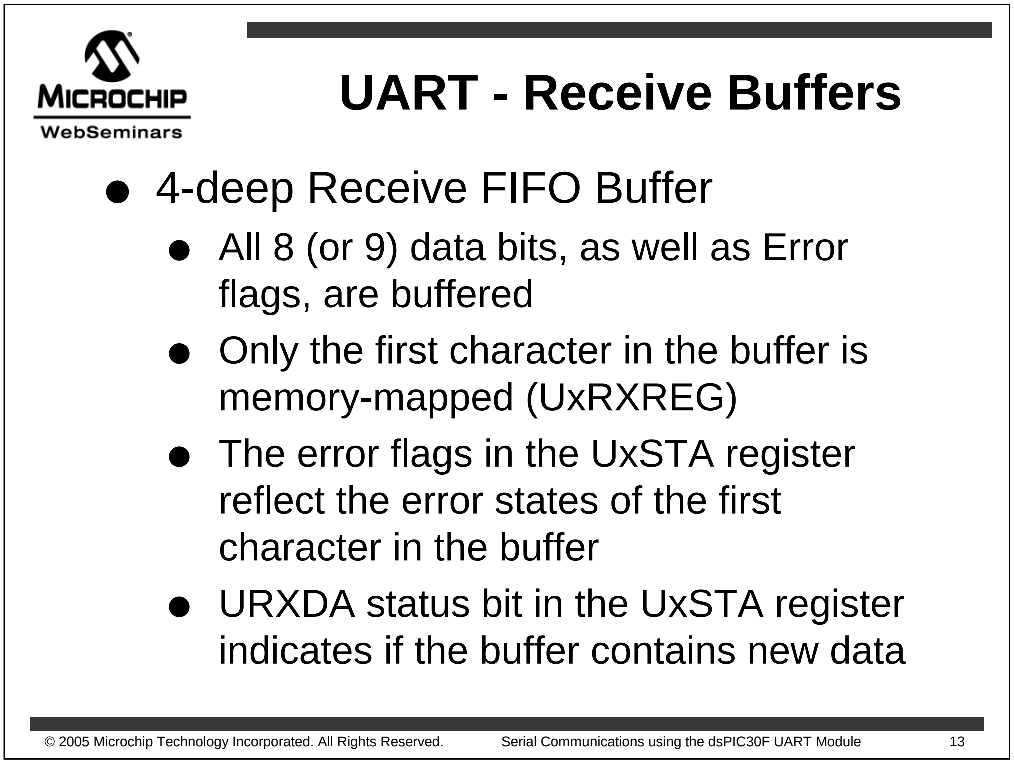
- UxRSR stops shifting in bits when it detects a STOP bit
  - Number of STOP bits (STSEL bit in the UxMODE register) must be configured to match those of the transmitting device
- As long as UxRSR is shifting in bits, the module sets the RIDLE status bit in the UxSTA register
  - Bit remains clear at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

© 2005 Microchip Technology Incorporated. All Rights Reserved. Serial Communications using the dsPIC30F UART Module 12

When the receiving device detects the required number of STOP bits, it stops shifting in data, and transfers the data from the Receive Shift Register to the Receive Buffer.

As is the case with the data format and parity, the receiving device must be configured for the same number of STOP bits as the transmitting device.

When the Receiver Idle status bit in the UART Status Register is clear, it indicates that a reception is in progress. When the Receiver is idle, then this bit is automatically set by the hardware.



MICROCHIP
WebSeminars

UART - Receive Buffers

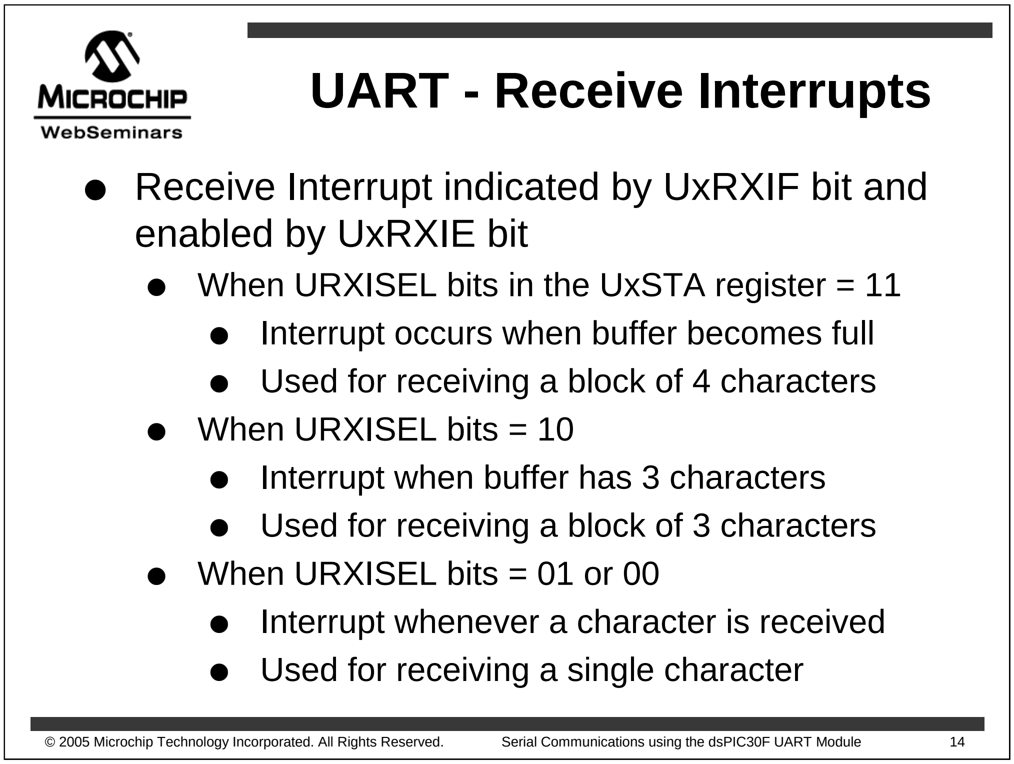
- 4-deep Receive FIFO Buffer
  - All 8 (or 9) data bits, as well as Error flags, are buffered
  - Only the first character in the buffer is memory-mapped (UxRXREG)
  - The error flags in the UxSTA register reflect the error states of the first character in the buffer
  - URXDA status bit in the UxSTA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

© 2005 Microchip Technology Incorporated. All Rights Reserved. Serial Communications using the dsPIC30F UART Module 13

As mentioned earlier, the Receive Buffer is a 4-level deep FIFO buffer. When a character is read from the Receive Buffer by the user, the remaining characters are automatically moved up one level within the buffer, thereby ensuring that the first buffer location always contains the oldest character in the buffer and the characters are read by the user in the same sequence in which they were received in the Receive Shift Register.

The error states are buffered along with the data character. The current state of the error flags in the UART Status Register always mirror the error status of the oldest character in the Receive Buffer.

The UART Receive Buffer Data Available status bit in the UART Status Register indicates whether the Receive Buffer contains any unread 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

Serial Communications using the dsPIC30F UART Module 14

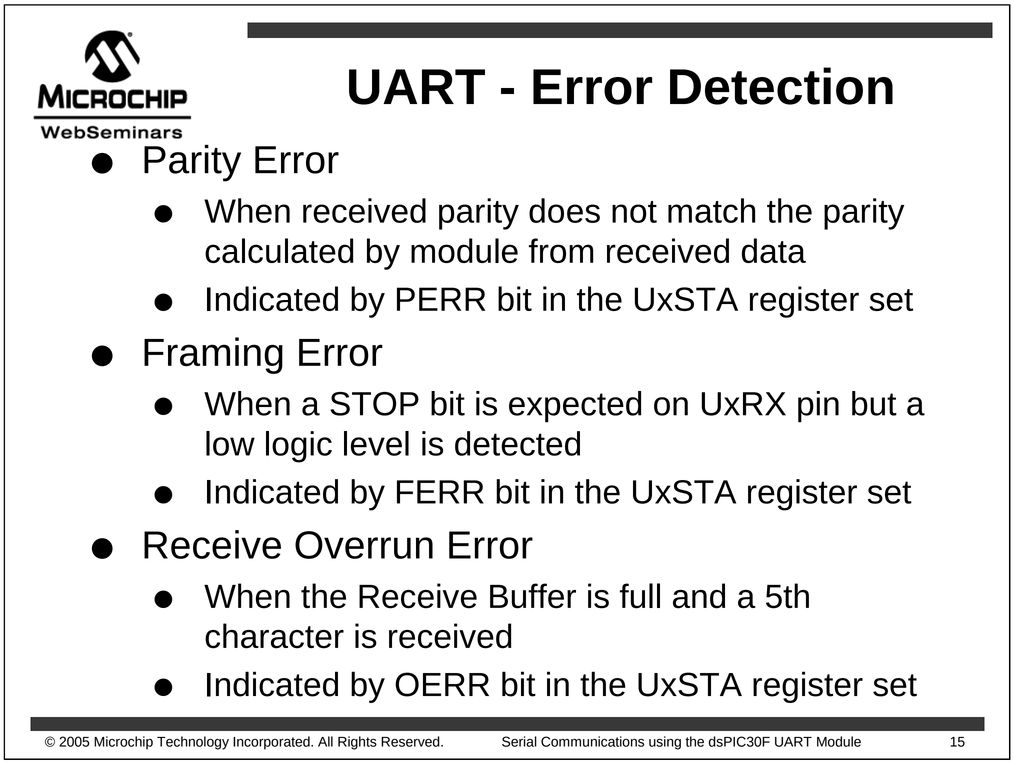
The UART module can generate a UART Receive Interrupt, which will be serviced if the corresponding interrupt enable bit is set and the interrupt priority is higher than the current CPU interrupt priority.

There are three Receive Interrupt modes:

If the UART Receive Interrupt Mode Select bits are set to the value ‘11’, then an interrupt is generated when a character transfer from the Receive Shift Register to the Receive Buffer causes the Receive Buffer to become full. This mode is useful for receiving a block of 4 characters, especially in applications in which generating an interrupt for reading each character may not be desirable.

If the UART Receive Interrupt Mode Select bits are set to the value ‘10’, then an interrupt is generated when a character is transferred from the Receive Shift Register to the Receive Buffer and the Receive Buffer contains 3 characters as a result. This mode is useful for receiving a block of 3 characters, again in applications in which generating an interrupt for reading each character may not be desirable.

If the UART Receive Interrupt Mode Select bits are ‘01’ or ‘00’, then an interrupt is generated every time a character is transferred from the Receive Shift Register to the Receive Buffer. It may be inferred that this mode is useful when the application can tolerate the overhead of servicing an interrupt for each character received.



MICROCHIP
WebSeminars

UART - Error Detection

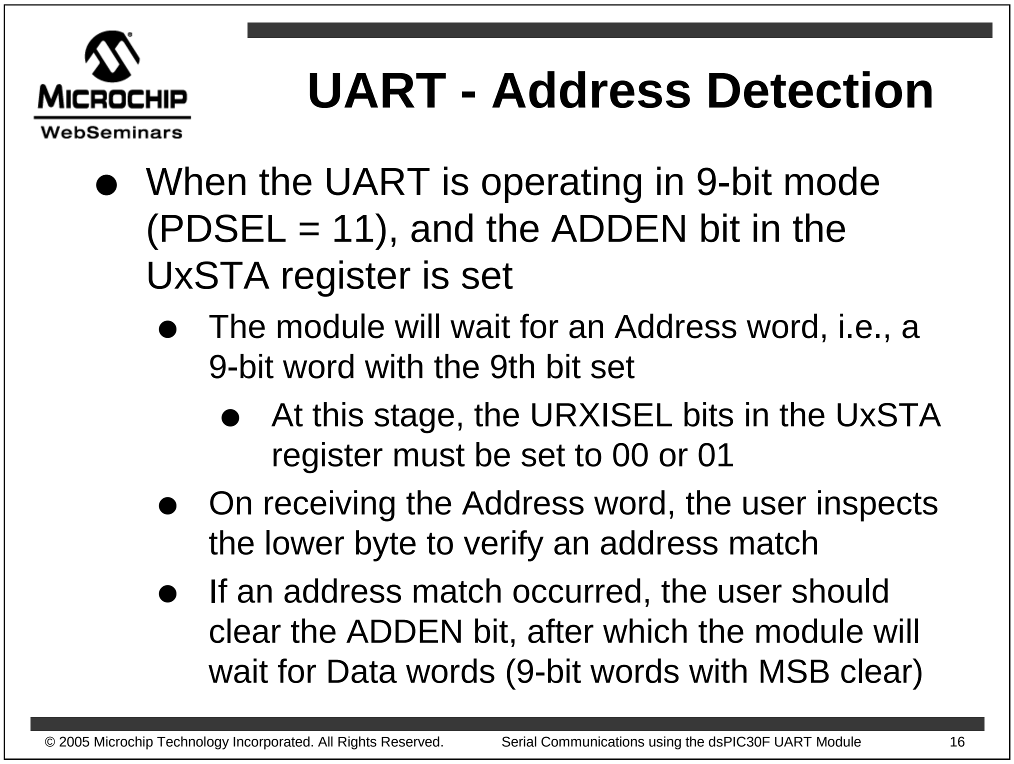
- Parity Error
  - When received parity does not match the parity calculated by module from received data
  - Indicated by PERR bit in the UxSTA register set
- Framing Error
  - When a STOP bit is expected on UxRX pin but a low logic level is detected
  - Indicated by FERR bit in the UxSTA register set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

The UART module has the capability of detecting various communication error conditions.

On receiving a character, the UART hardware calculates the parity bit for the received data bits. If this calculated parity does not match the received parity bit, then a Parity Error is signaled by the hardware, setting the Parity Error bit in the UART Status Register.

When a UART has received the required number of data bits, the receiver hardware expects the specified number of STOP bits on its Receive line. If it detects a low level on the Receive pin when it expected a STOP bit, a Framing Error is signaled by the hardware, setting the Framing Error bit in the UART Status Register.

A Receive Overrun condition occurs when the Receive Buffer is full and a new character is completely shifted into the Receive Shift Register. This error is signaled by the hardware, setting the Receive Overrun Error bit in the UART Status Register. The newly received character is discarded.



UART - Address Detection

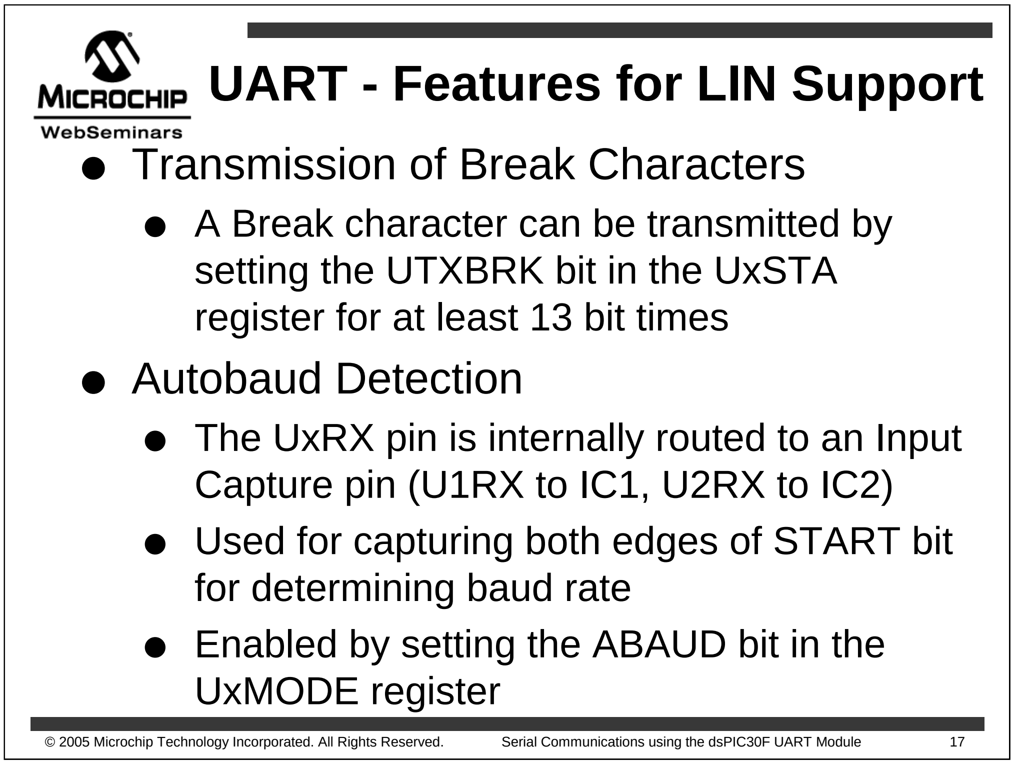
- When the UART is operating in 9-bit mode (PDSEL = 11), and the ADDEN bit in the UxSTA register is set
  - The module will wait for an Address word, i.e., a 9-bit word with the 9th bit set
    - At this stage, the URXISEL bits in the UxSTA register must be set to 00 or 01
  - On receiving the Address 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)

Serial Communications using the dsPIC30F UART Module 16

If the data size is set to 9 bits, the UART module also supports a special feature known as Address Detection. This mode is enabled by setting the Address Detect Enable bit in the UART Status Register.

If the Address Detect Enable bit is set, and if the UART Receive Interrupt Mode Select bits are set to ‘00’ or ‘01’, then the module waits for a 9-bit character with the MSB set. As soon as it receives such a character, it interprets the remaining 8 bits to be an Address. The user software can then inspect this 8-bit address to ascertain if it is the device being addressed.

On a successful Address match, the user should clear the Address Detect Enable bit, and if needed, re-configure the UART Receive Interrupt Mode Select bits. Subsequent characters with the MSB cleared are interpreted as data characters.



MICROCHIP WebSeminars

UART - Features for 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
- Autobaud Detection
  - The UxRX pin is internally routed to an Input Capture pin (U1RX to IC1, U2RX to IC2)
  - Used for capturing both edges of START bit for determining baud rate
  - Enabled by setting the ABAUD bit in the UxMODE register

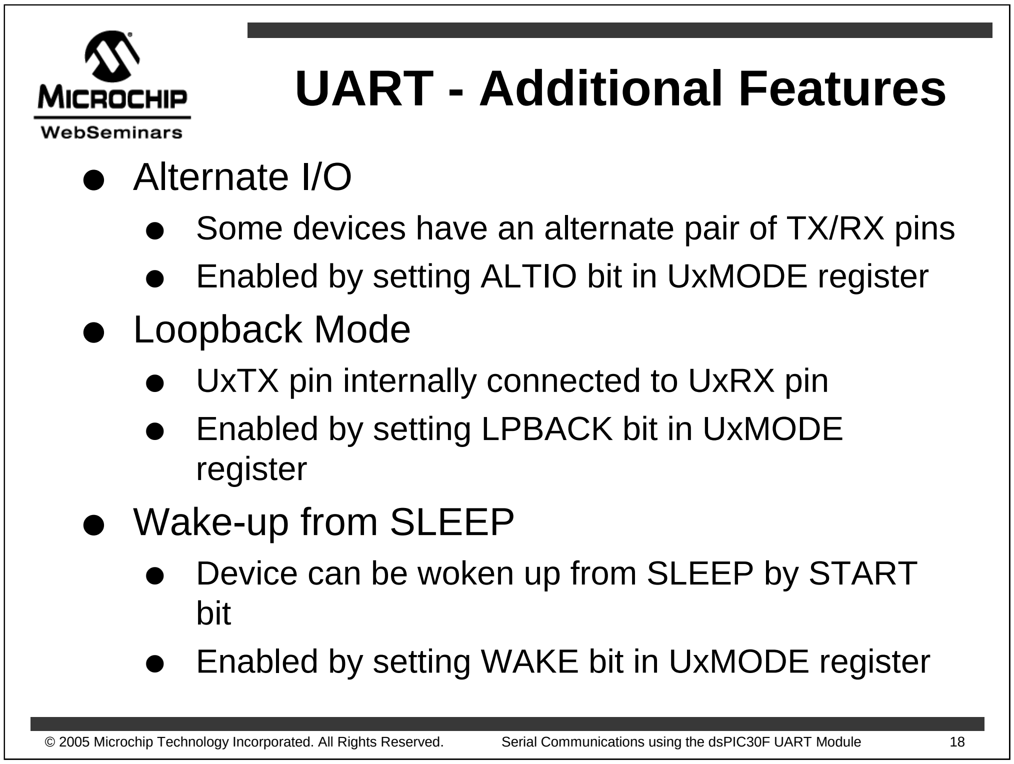
© 2005 Microchip Technology Incorporated. All Rights Reserved. Serial Communications using the dsPIC30F UART Module 17

The UART module has some key features that primarily support the LIN protocol, but can also be used by user applications for other purposes.

Setting the UART Transmit Break bit, and keeping it set for a minimum of 13 bit times, generates a ‘zero’ character on the Transmit pin. In other words, the Transmit pin is held low as long as the UART Transmit Break bit is set. A STOP bit can then be generated by clearing the UART Transmit Break bit. If the receiving device does not receive a STOP bit, then it generates a Framing Error. Unlike other characters, a Break character does not generate a Transmit Interrupt.

Another feature that supports the LIN protocol is Autobaud Detection. This refers to the capability of the UART to internally route the Receive signal to a specified Input Capture pin. The corresponding Input Capture channel, IC1 for UART1 and IC2 for UART2, should be configured to capture every edge. Now, if a START bit is received, the Capture values of its falling and rising edge can be utilized by user software to determine the current Baud Rate of communications on the bus. This feature is enabled by setting the Autobaud bit in the UART Mode Register.

Application Note AN962 titled “Implementing Autobaud on dsPIC30F Devices” further explains this feature.



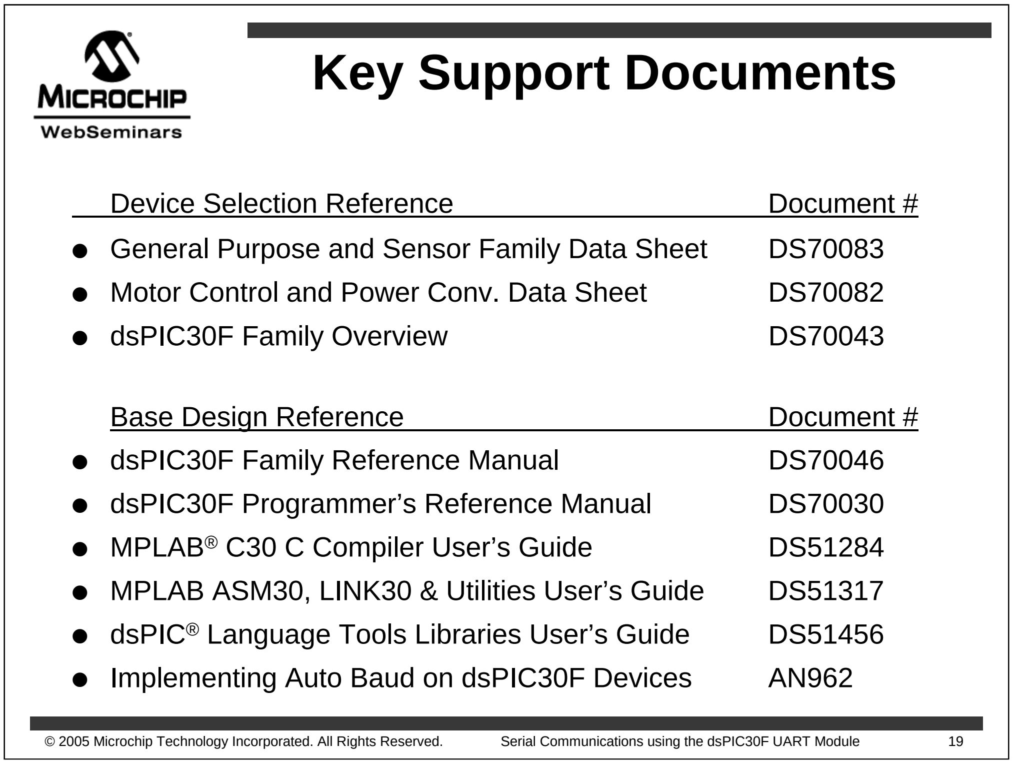
UART - Additional Features

- Alternate I/O
  - Some devices have an alternate pair of TX/RX pins
  - Enabled by setting ALTIO bit in UxMODE register
- Loopback Mode
  - UxTX pin internally connected to UxRX pin
  - Enabled by setting LPBACK bit in UxMODE register
- Wake-up from SLEEP
  - Device can be woken up from SLEEP by START bit
  - Enabled by setting WAKE bit in UxMODE register

Some dsPIC30F devices contain an alternate pair of UART Transmit and Receive pins. If the user sets the Alternate I/O Enable bit in the UART Mode Register, then these alternate pins are used instead of the normal UART Transmit and Receive pins.

For diagnostic purposes, the UART also supports an internal Loopback Mode, wherein the UART Transmit signal is internally routed to the UART Receive signal, and the external UART Transmit and Receive pins are not used.

Last but not the least, a device in SLEEP mode can optionally be woken up on detection of a START bit on its UART Receive pin. This feature is enabled by setting the Wake-up Enable bit in the UART Mode Register. If the CPU is not performing any tasks other than waiting to receive characters through the UART, then putting the device into SLEEP mode, and waking it up only when needed, can reduce the overall power consumption of the device.



Key Support Documents

Device Selection Reference	Document #
● General Purpose and Sensor Family Data Sheet	DS70083
● Motor Control and Power Conv. Data Sheet	DS70082
● dsPIC30F Family Overview	DS70043

Base Design Reference	Document #
● dsPIC30F Family Reference Manual	DS70046
● dsPIC30F Programmer’s Reference Manual	DS70030
● MPLAB® C30 C Compiler User’s Guide	DS51284
● MPLAB ASM30, LINK30 & Utilities User’s Guide	DS51317
● dsPIC® Language Tools Libraries User’s Guide	DS51456
● Implementing Auto Baud on dsPIC30F Devices	AN962

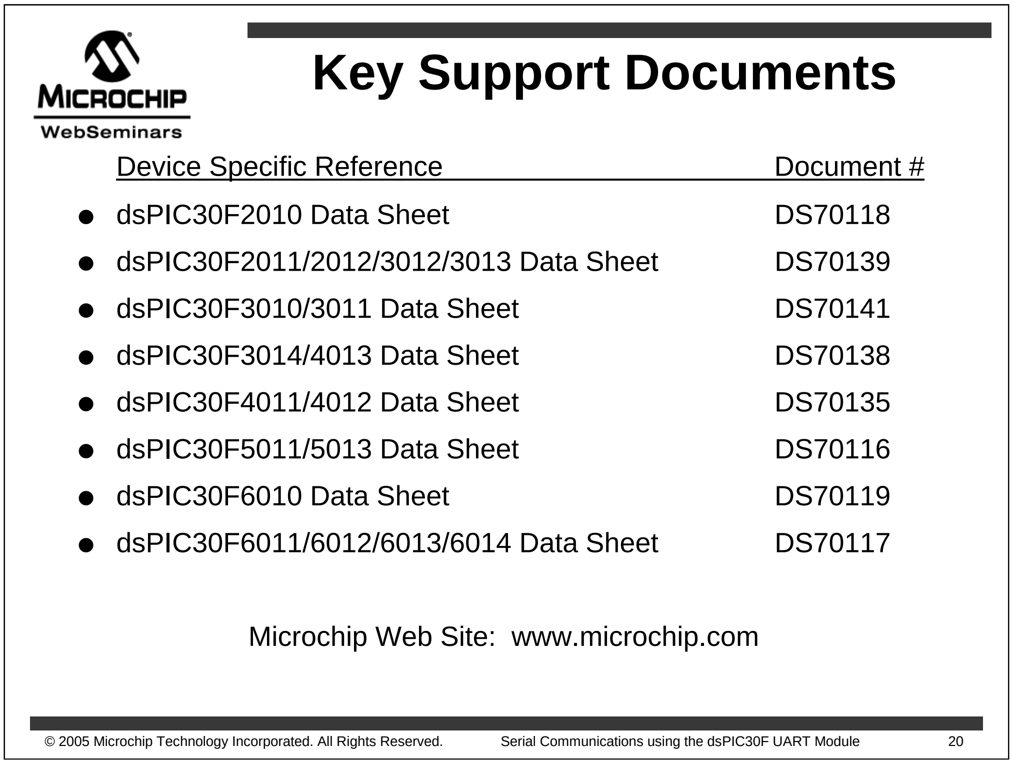
© 2005 Microchip Technology Incorporated. All Rights Reserved. Serial Communications using the dsPIC30F UART Module 19

Well this wraps up today’s session on the UART module. We hope you found the presentation useful and informative.

For more information, here are references to some important documents that contain a wealth of information about the dsPIC30F family of devices.

The Family Reference Manual contains detailed information about the architecture and peripherals, whereas the Programmer’s Reference Manual contains a thorough description of the instruction set.

Don’t forget that application note AN962 explains how to use Auto Baud on the dsPIC30F.



Key Support Documents

Device Specific Reference	Document #
● dsPIC30F2010 Data Sheet	DS70118
● dsPIC30F2011/2012/3012/3013 Data Sheet	DS70139
● dsPIC30F3010/3011 Data Sheet	DS70141
● dsPIC30F3014/4013 Data Sheet	DS70138
● dsPIC30F4011/4012 Data Sheet	DS70135
● dsPIC30F5011/5013 Data Sheet	DS70116
● dsPIC30F6010 Data Sheet	DS70119
● dsPIC30F6011/6012/6013/6014 Data Sheet	DS70117

Microchip Web Site: www.microchip.com

For device-specific information such as pinout diagrams, packaging and electrical characteristics, the device datasheets listed here are the best source of information.

All these documents can be obtained from the Microchip web site shown, by clicking on the “dsPIC® Digital Signal Controllers” or “Technical Documentation” link.

Thanks again for attending today’s training session and your interest in the dsPIC30F family of Digital Signal Controllers.