11037 SER

Communicate with PIC16 and PIC18 MCUs via the PC Serial Port

UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

Class Objectives

When you walk out, you will know:

- **Benefits of the PC serial interface**
- **PC to PIC® microcontroller (MCU) hardware interface**
- **PIC MCU USART peripherals**
- **PC Serial interface applications**
  - Built-in Programming Interface Functions
    - C#
    - Visual Basic

UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

Class Agenda

- **Overview of PC to PIC® MCU Communications Solution**
- **PC to PIC MCU hardware interface**
- **Setup and use of the USART peripheral**
- **Communications Firmware**
- **Serial port methods and properties**
- **PC Software**
  - C#
  - Visual Basic

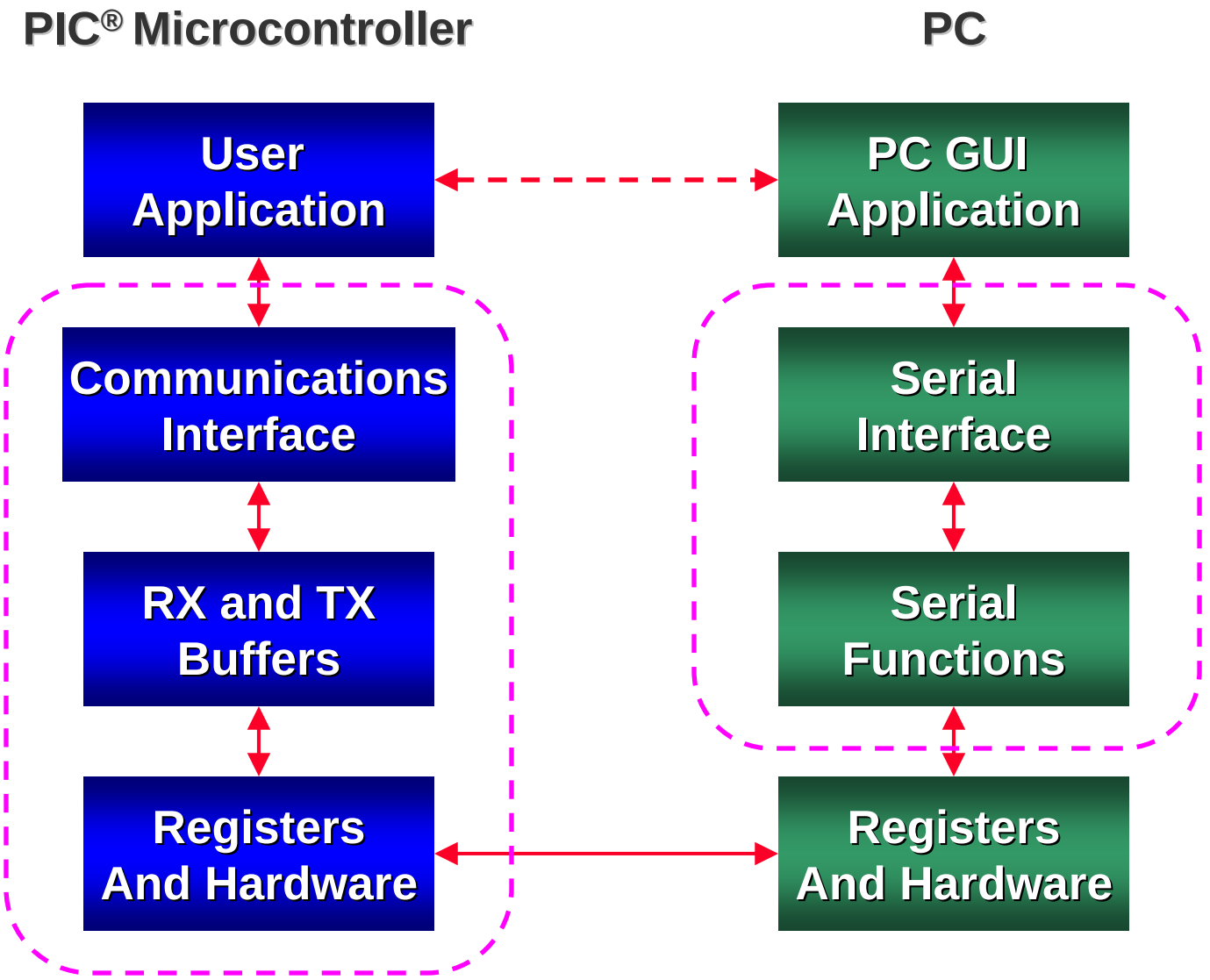
UNIVERSITY OF MICROCHIP
UOFM
MASTERS 2007

Serial Communications Overview

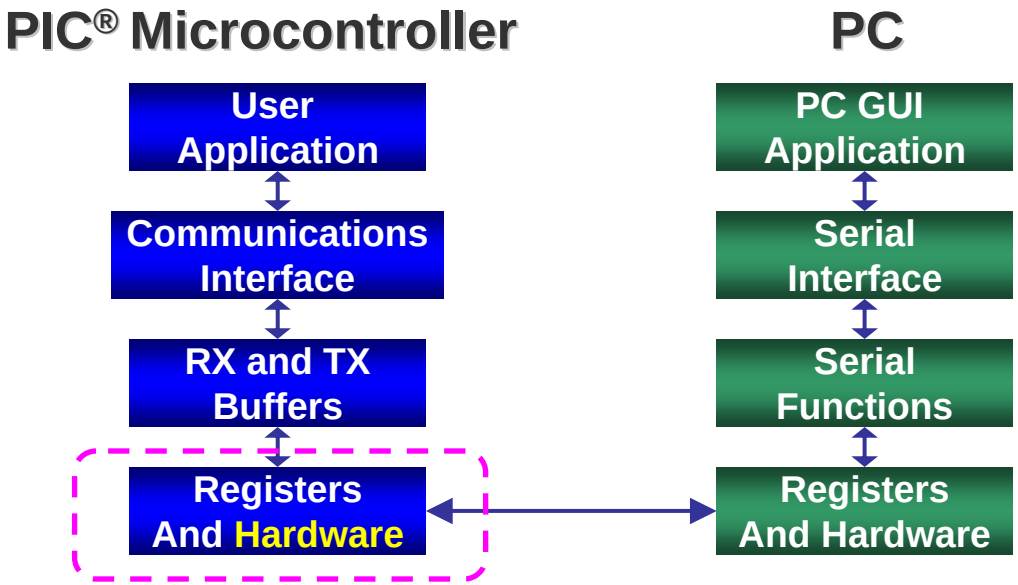
Benefits of Serial

- Universal
- Simple
- Some implementations require only 1 wire
- Easy to troubleshoot
- PCs supplied with Hyperterminal
- No Vendor ID required
- Expandable with USB-to-Serial adapters

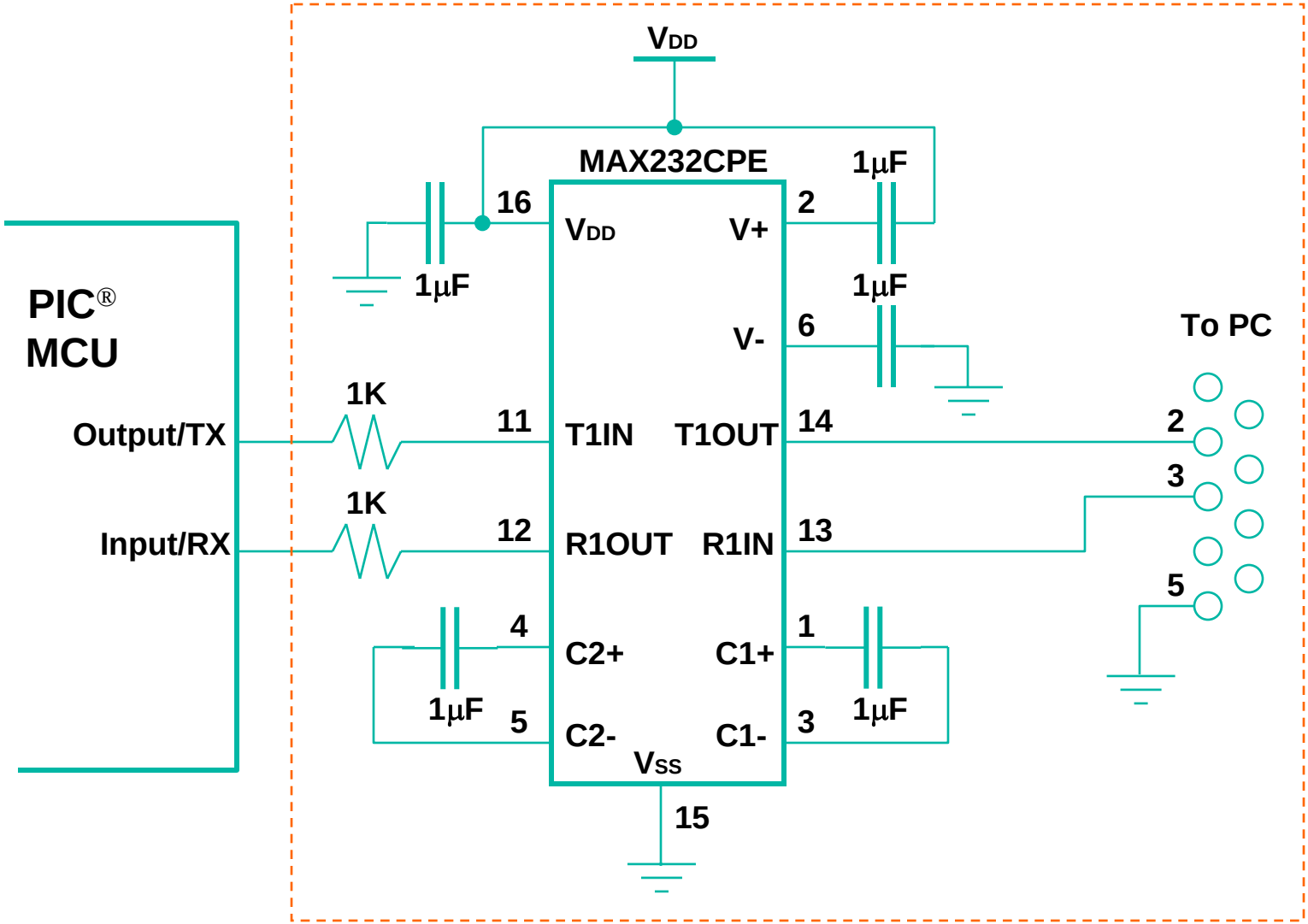
Serial Solution Layers



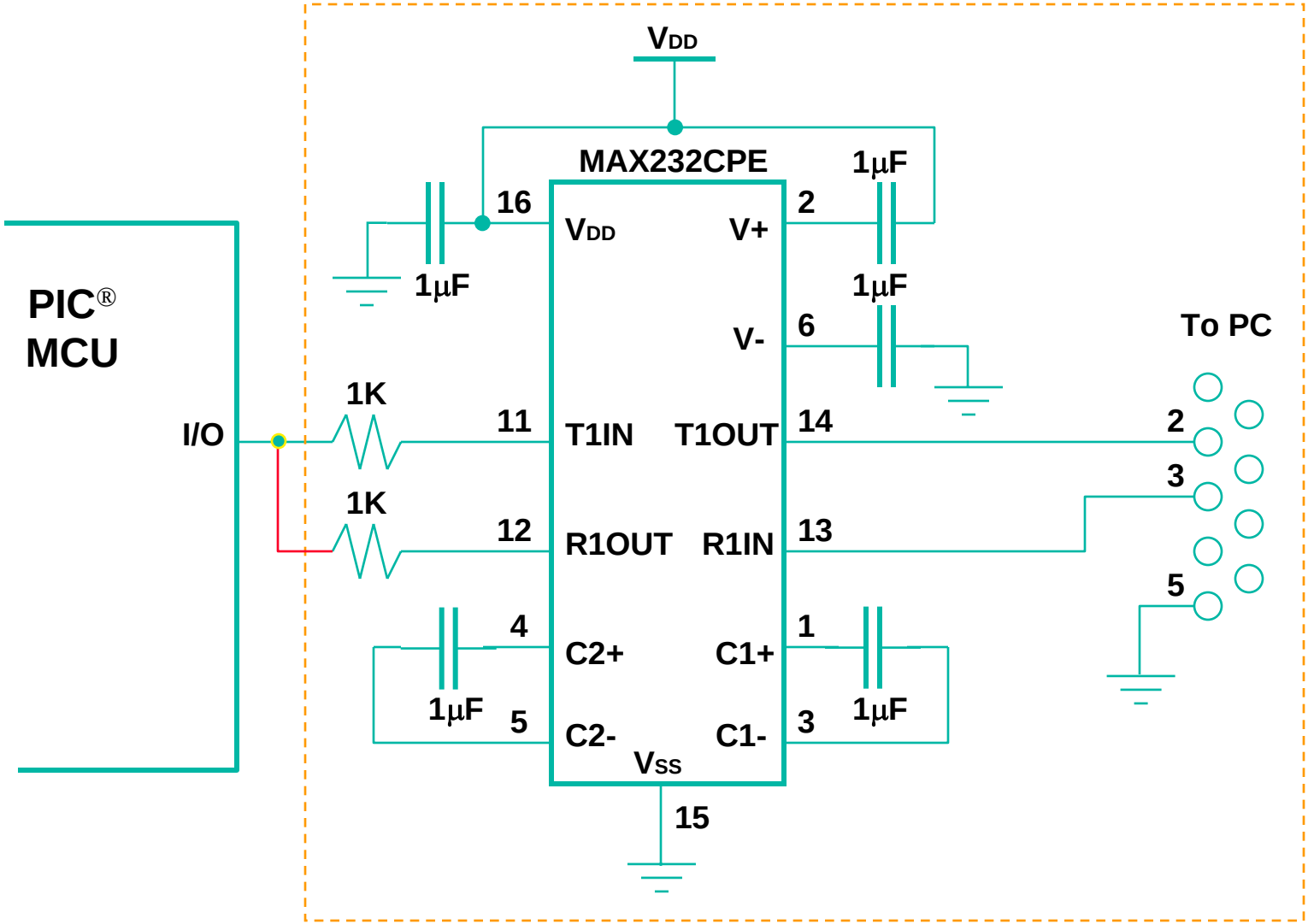
Hardware



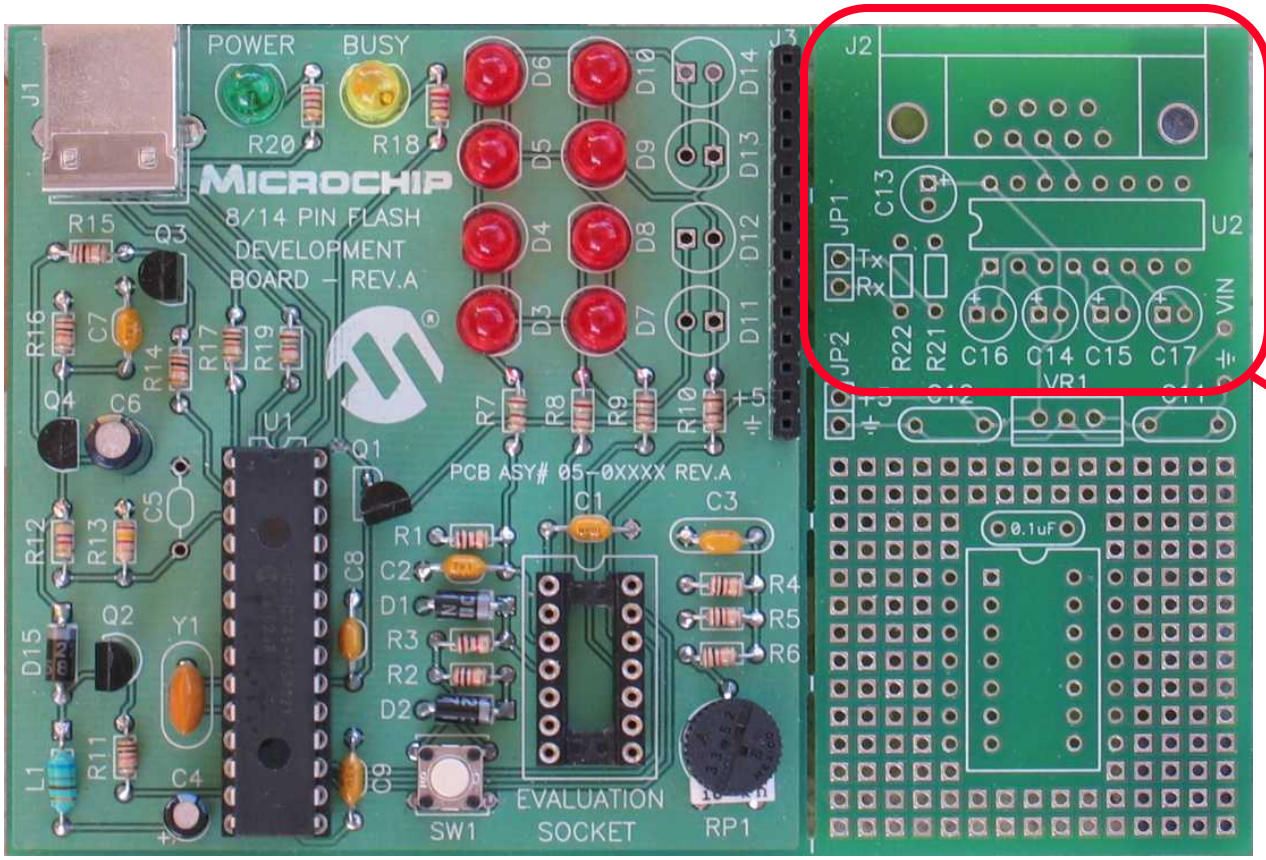
Typical 2 Wire PC Interface



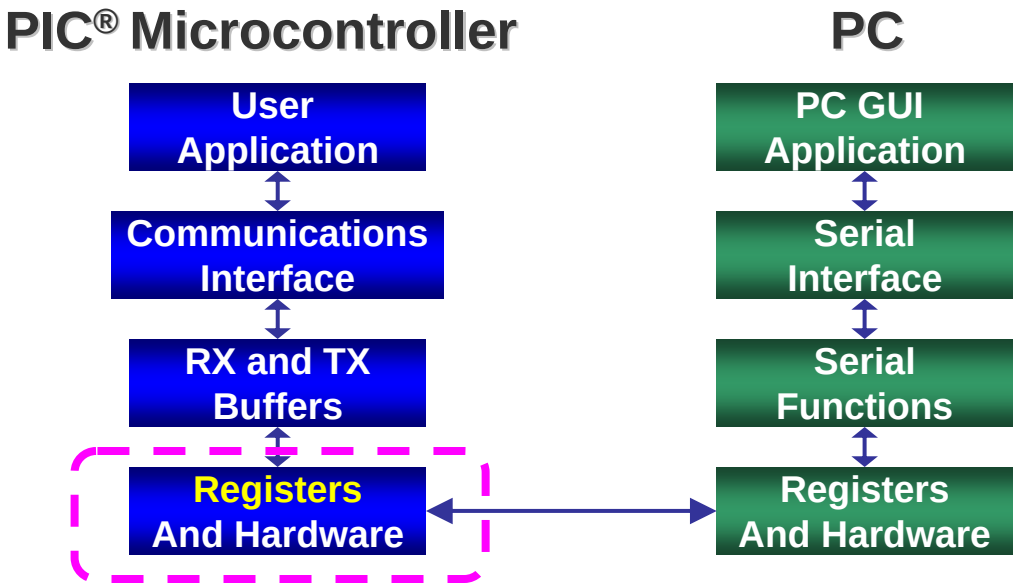
Typical 1 Wire PC Interface



PICkit™ Starter Kit Circuit Board



USART Peripheral



USART Types

- **USART**
  - Universal Synchronous/Asynchronous Receiver Transmitter
- **AUSART**
  - Same as USART with address recognition
- **EUSART**
  - Enhanced - Same as AUSART with
    - Baud rate detection/calibration
    - Wake on break
    - Break character transmit

USART Registers

- RCSTA – Receive status and control
- TXSTA – Transmit status and control
- RCREG – Receive data
- TXREG – Transmit data
- SPBRG – 8 bit baud rate selection
- PIR and PIE – Interrupt status and enable
- SPBRGH – upper 8 bits (EUSART only)
- BAUDCTL – Extended control (EUSART only)

Receive Status and Control

RCSTA

CONTROL

- SPEN – Serial peripheral enable (turns on the USART)
- CREN – Continuous receive enable (turns on the receiver)

STATUS

- FERR – Framing error (cleared by reading RCREG)
- OERR – Overrun error (cleared by clearing CREN)

NOT USED (in this application)

Transmit Status and Control

TXSTA

CONTROL

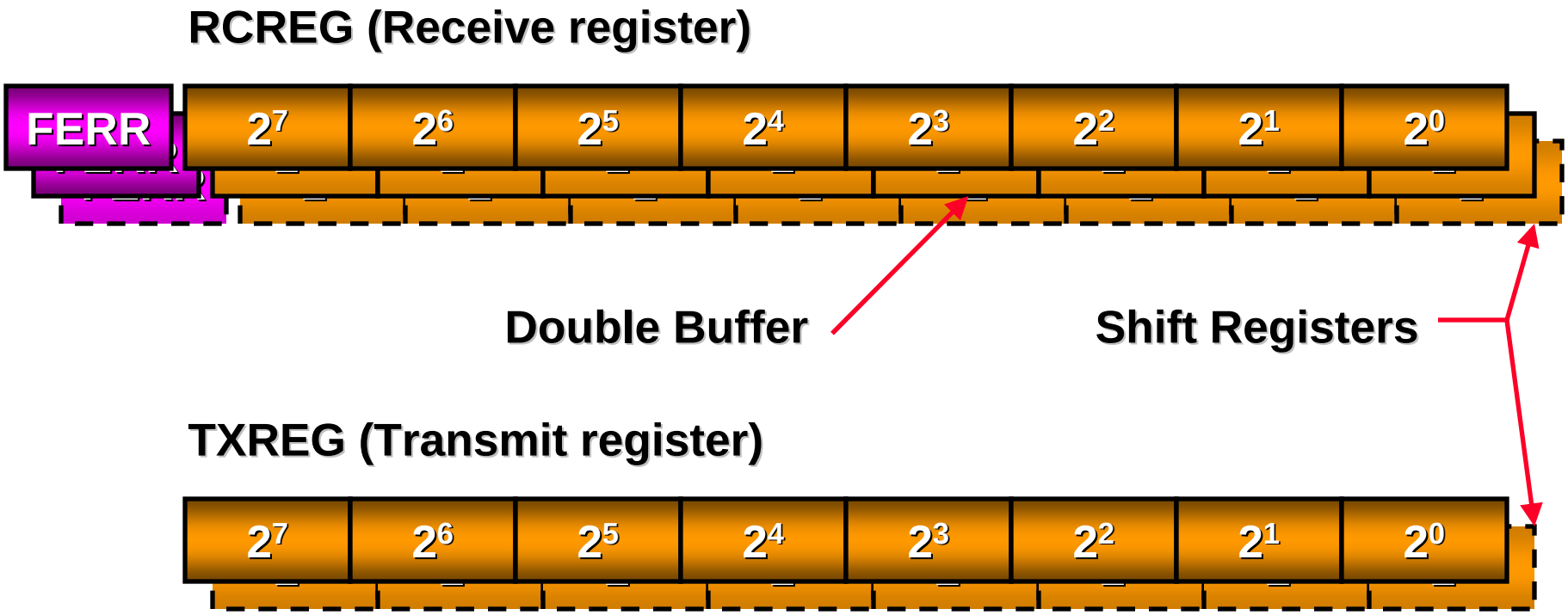
- TXEN – Transmit enable (turns on transmit)
- BRGH – High range baud rate (faster baud rate clock)

STATUS

- TRMT – Transmit shift register empty

NOT USED (in this application)

Serial Data Registers



Interrupt Status and Enable

PIR1

CONTROL

- **TXIF – TXREG is empty (read only)**
- **RCIF – RCREG is full (read only)**

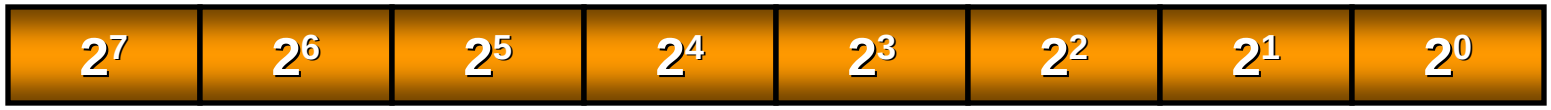
PIE1

CONTROL

- **TXIE – Enable TXIF interrupts**
- **RCIE – Enable RCIF interrupts**

Baud Rate Generator

SPBRG



CONTROL

SPBRG = Number of clocks in one bit period

SPBRG = [Fosc/(DataRate·Div)] -1

When BRGH = 0: Div = 64

When BRGH = 1: Div = 16

SPBRGH (EUSART only)

NOT USED (in this application)

Baud Rate Control

BAUDCTL (EUSART only)

CONTROL

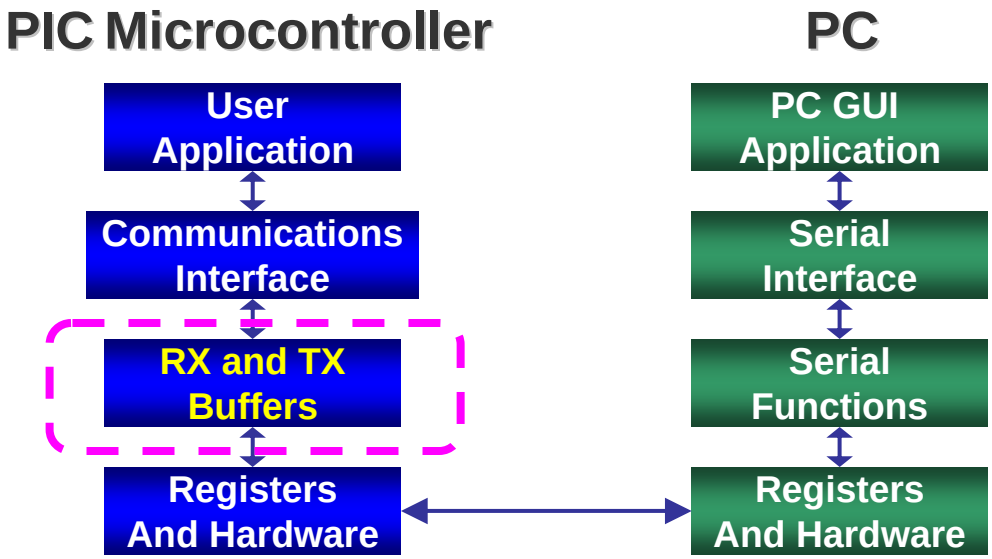
ABDEN = Autobaud detect enable

STATUS

ABDOVF = Autobaud detect overflow

NOT USED (in this application)

PIC® MCU Firmware Buffer Level



Receive and Transmit Buffering

- **Circular buffers**
  - All communications via buffer fill and empty routines
  - Allows back to back messaging
- **Receive buffering**
  - Entire command received before processing
  - Host PC doesn't need to wait
  - Permits typing corrections
- **Transmit buffering**
  - Allows multi-byte conversions in one step

University of Microchip UofM MASTERs 2007

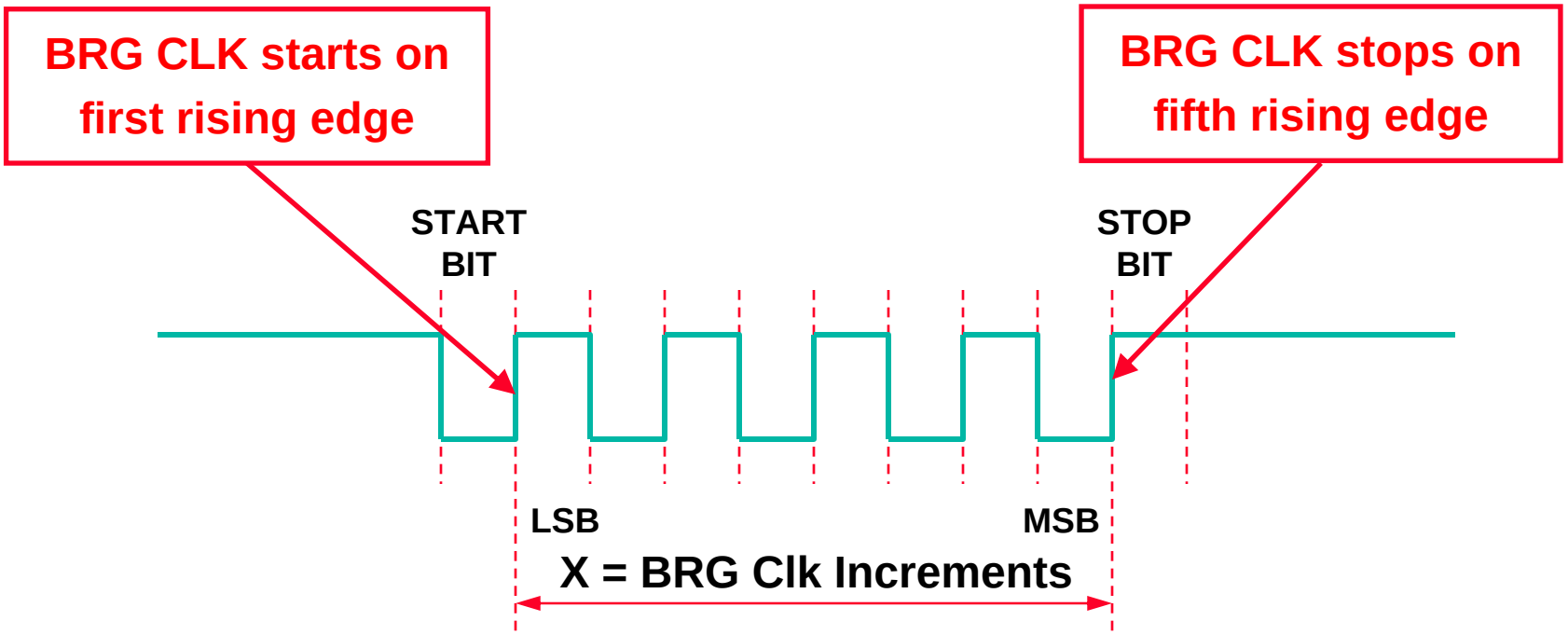
Buffer Functions

- **UARTInit:** Initialize SFRs and pointers
- **ResetRXPointers:** Set PutRX to GetRX
- **Autobaud:** Measure cal char and set SPBRG
- **TXService:** Transfer TX buffer to TXREG
- **RXService:** Transfer RCREG to RX buffer
- **GetRX:** Get character from RX buffer
- **PutTX:** Put character in TX buffer

Autobaud

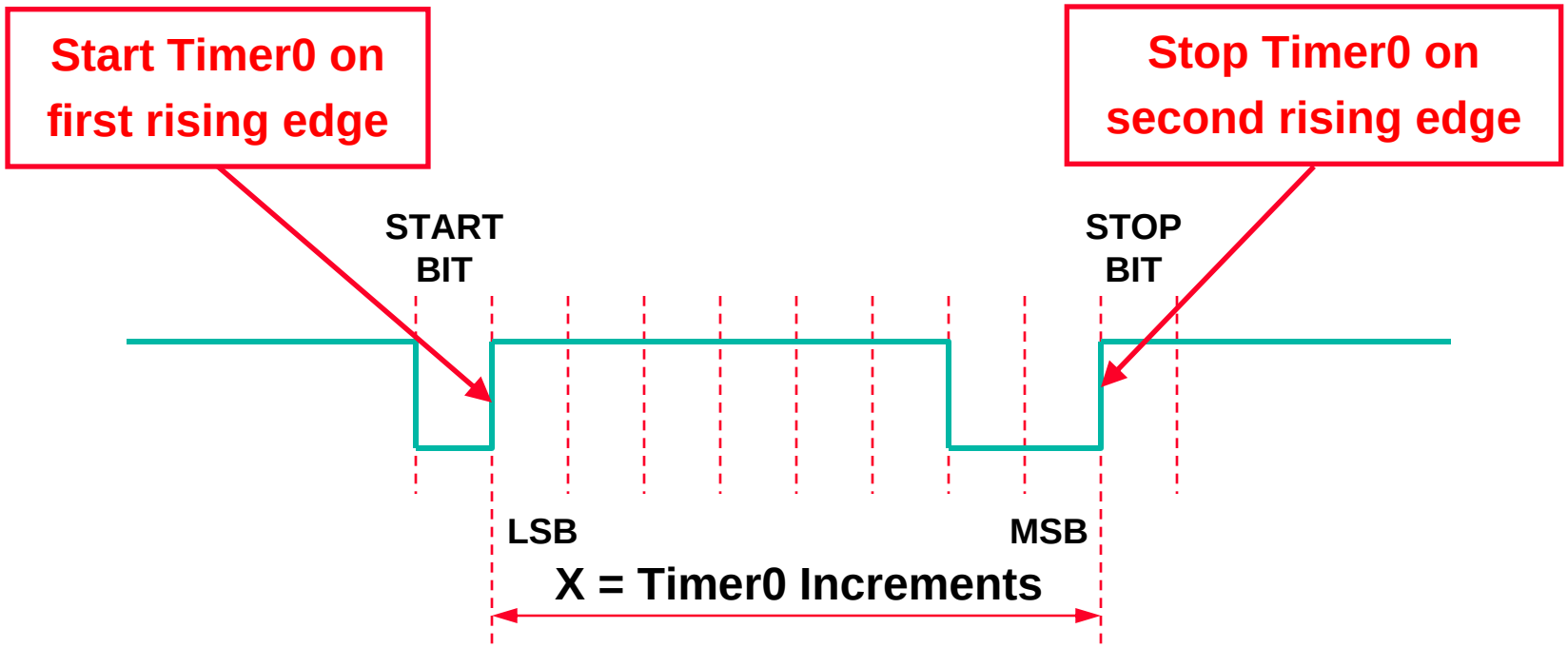
- **Adjust SPBRG based on calibration character**
- **Precludes the need for a crystal**
- **RC Oscillator**
  - Low cost solution
  - Frees I/O pins
- **LIN bus requirement**

EUSART Autobaud - Timing



- Only possible calibration character: 0x55 = “U”
- BRG Clock is 1/8 base BRG clock rate during AUTOCAL
- RCIF flag is set at completion. RCIF is cleared by reading RCREG

Autobaud - Timing



All possible calibration characters

0x01 = SOH
0x03 = ETX
0x07 = ACK
0x0F = SI
0x1F = US
0x3F = "?"
0x7F = DEL

University of Microchip UofM MASTERs 2007

Autobaud - Derivation

SPBRG = [(Timer0 Counts*Timer0Prescale)/(4*Bits)] -1

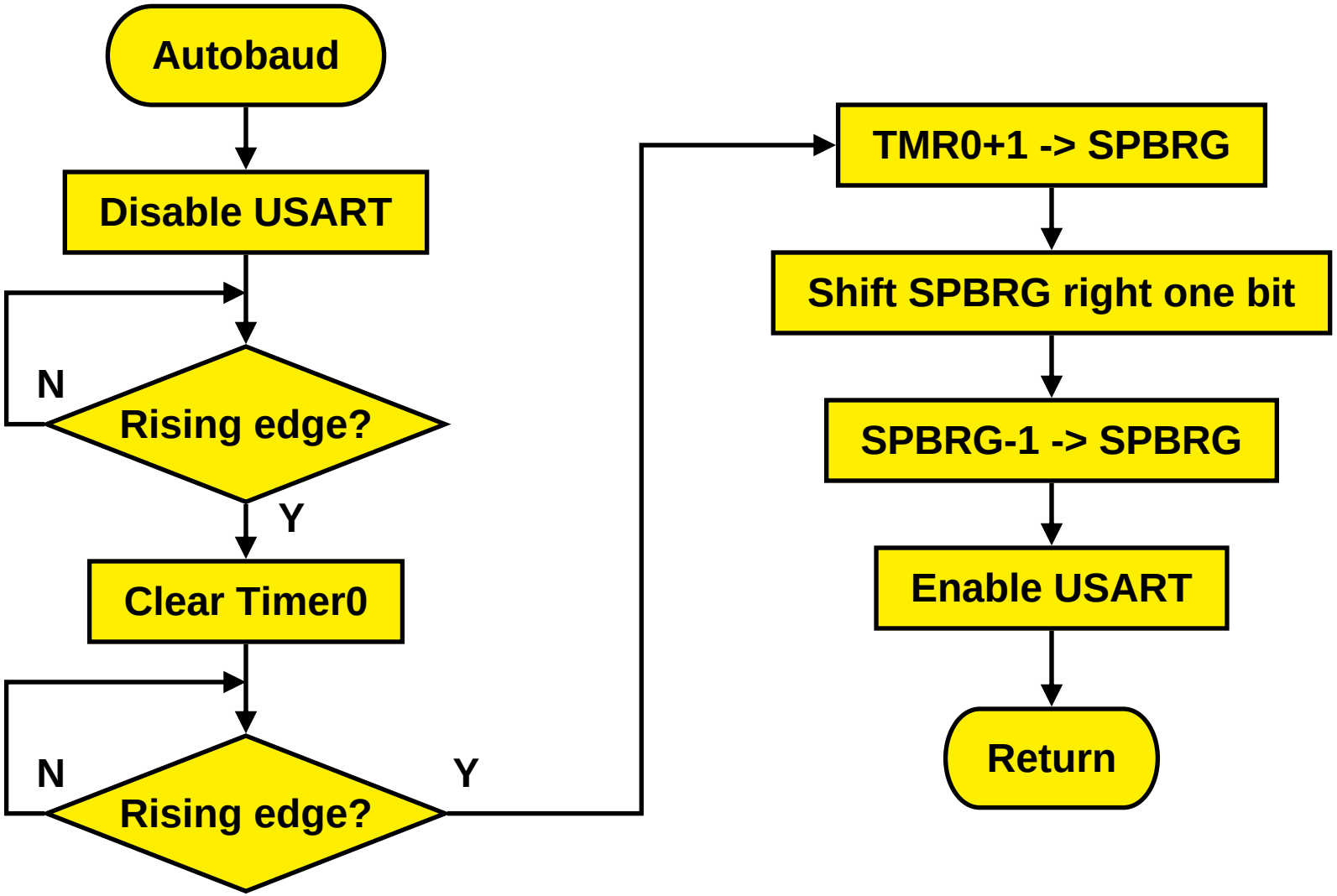
Assuming Timer0 Prescale = 2*Bits

SPBRG = (Timer0 Counts/2) -1

(X/2 allows rounding up resulting in an uncertainty of +/- 0.5 counts)

Full derivation can be found in code listings.

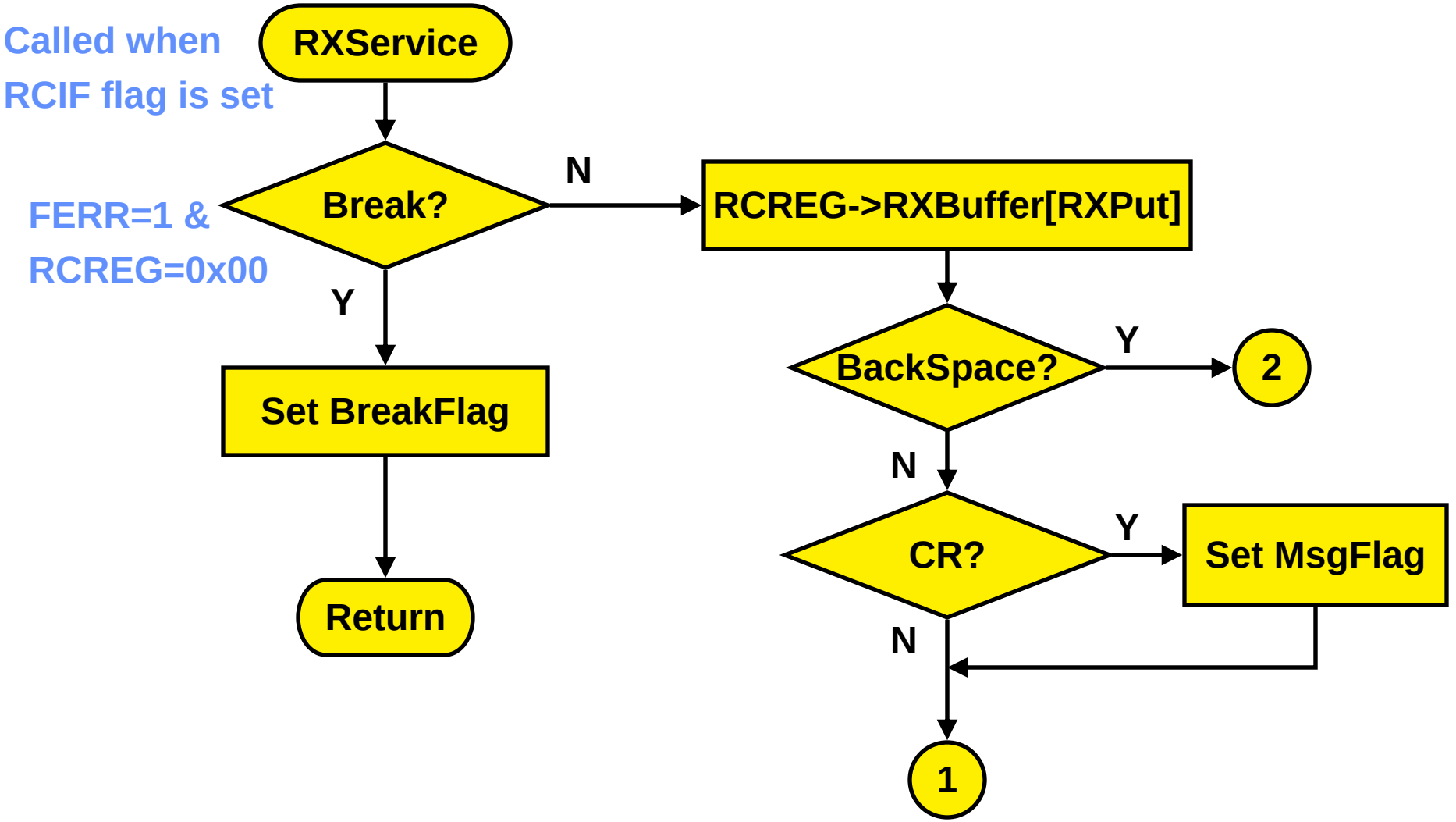
Autobaud



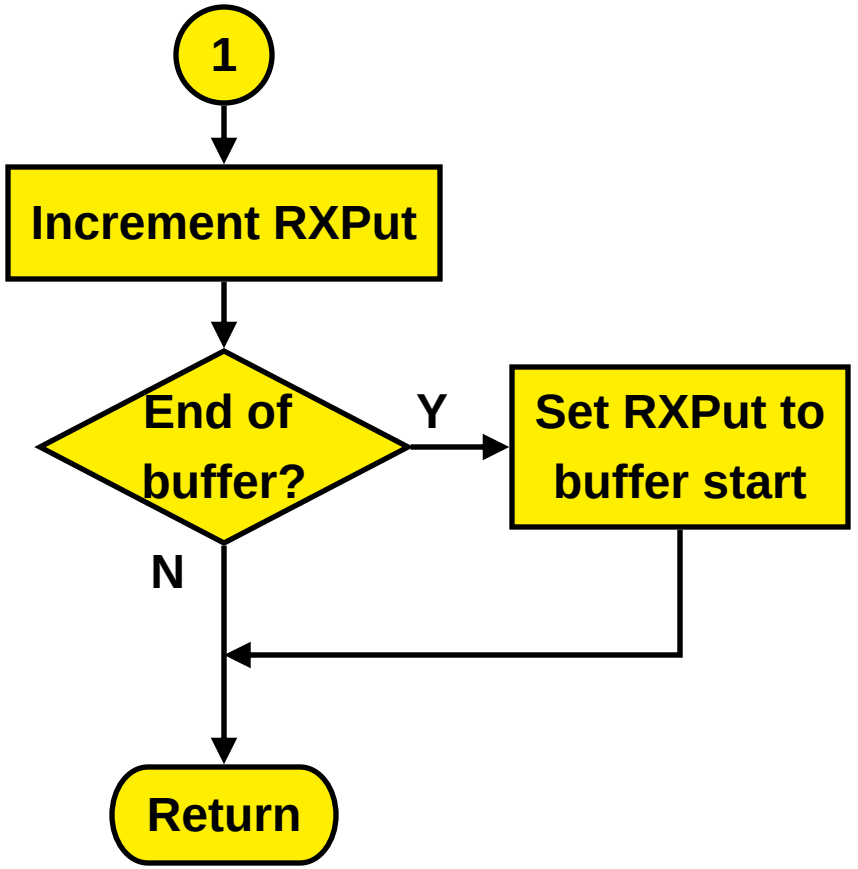
Receive Buffer

- **As large as longest command plus margin**
- **Two pointers**
  - RXPut
    - Put pointer for inserting bytes from RCREG
  - RXGet
    - Get pointer for removing bytes
- **Flags**
  - RCIF invokes RXService
  - GetMsg invokes GetRX
  - BreakFlag invokes recalibration

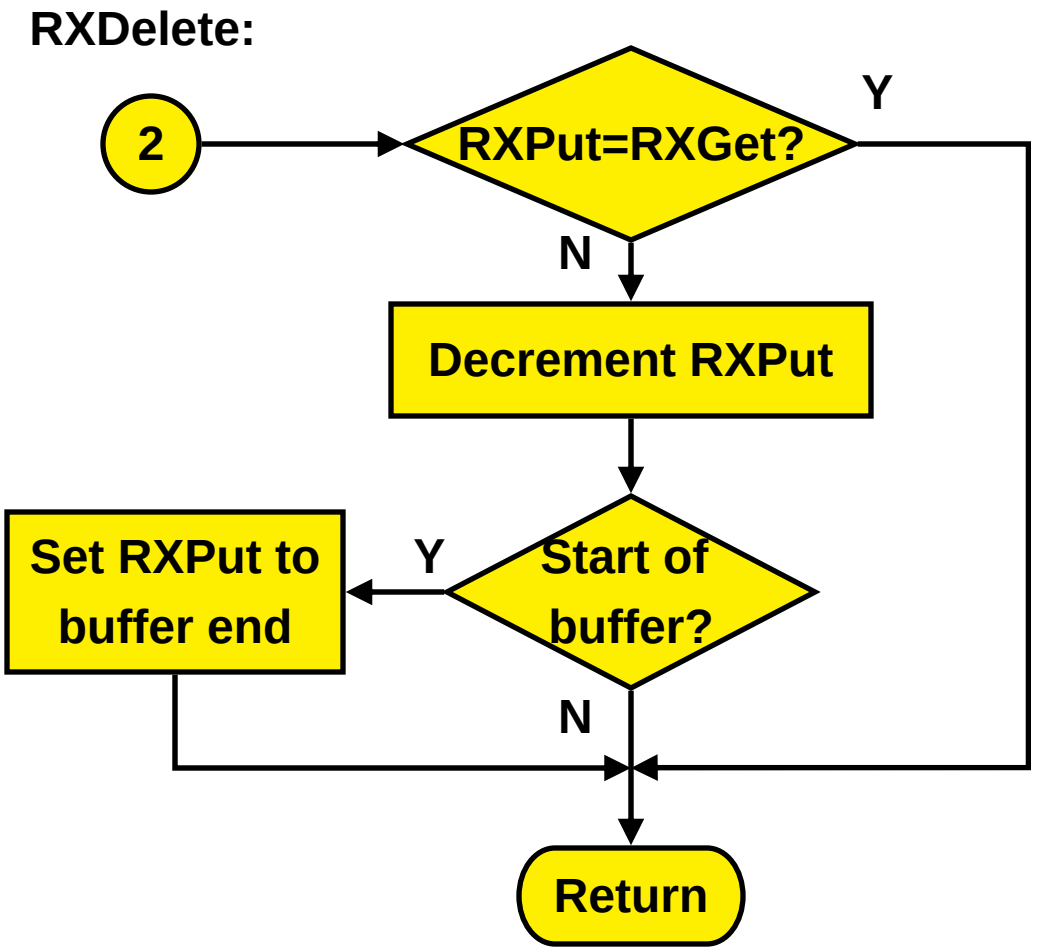
Fill Receive Buffer



Fill Receive Buffer (cont)

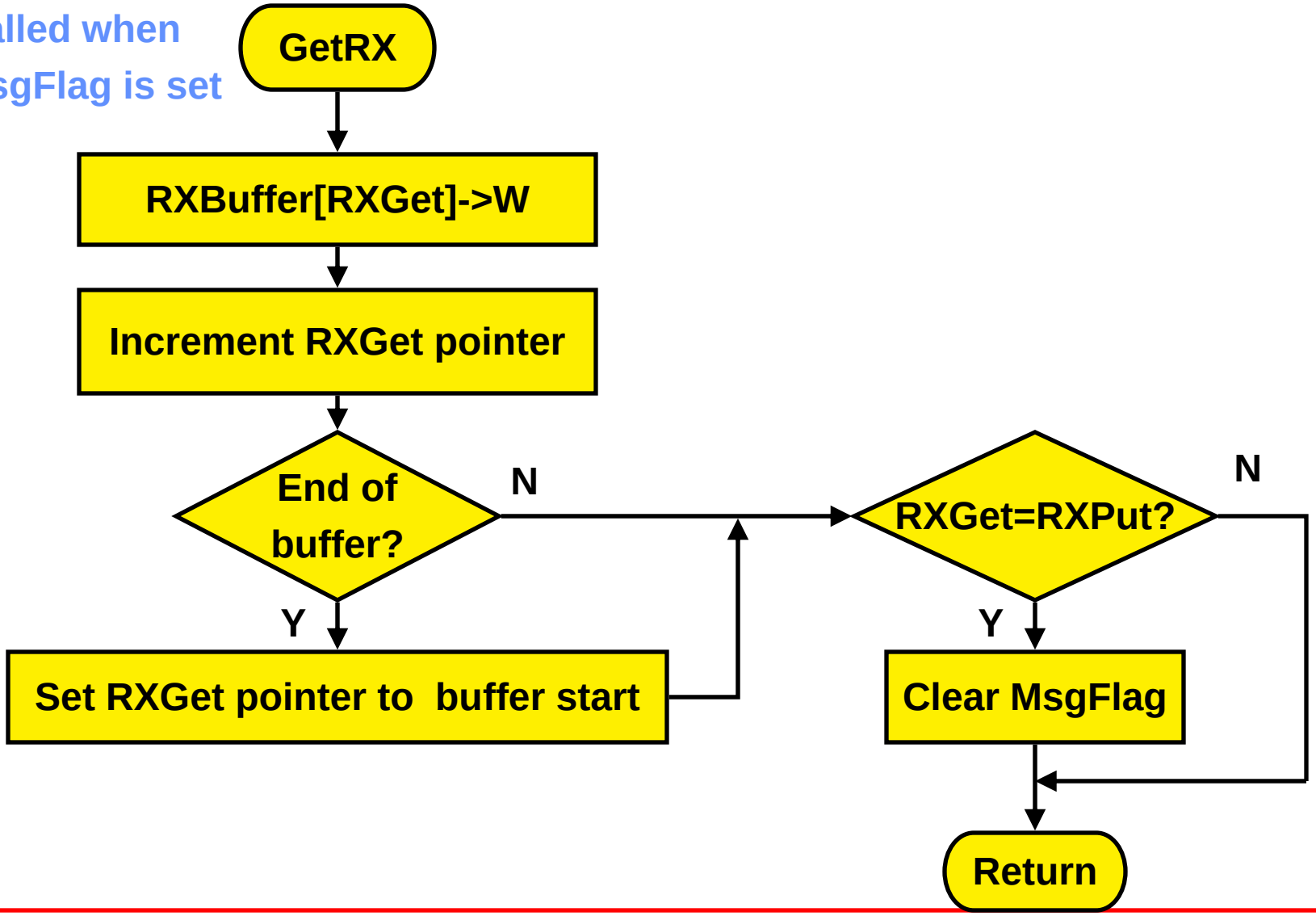


Fill Receive Buffer (cont)



Empty Receive Buffer

Called when MsgFlag is set

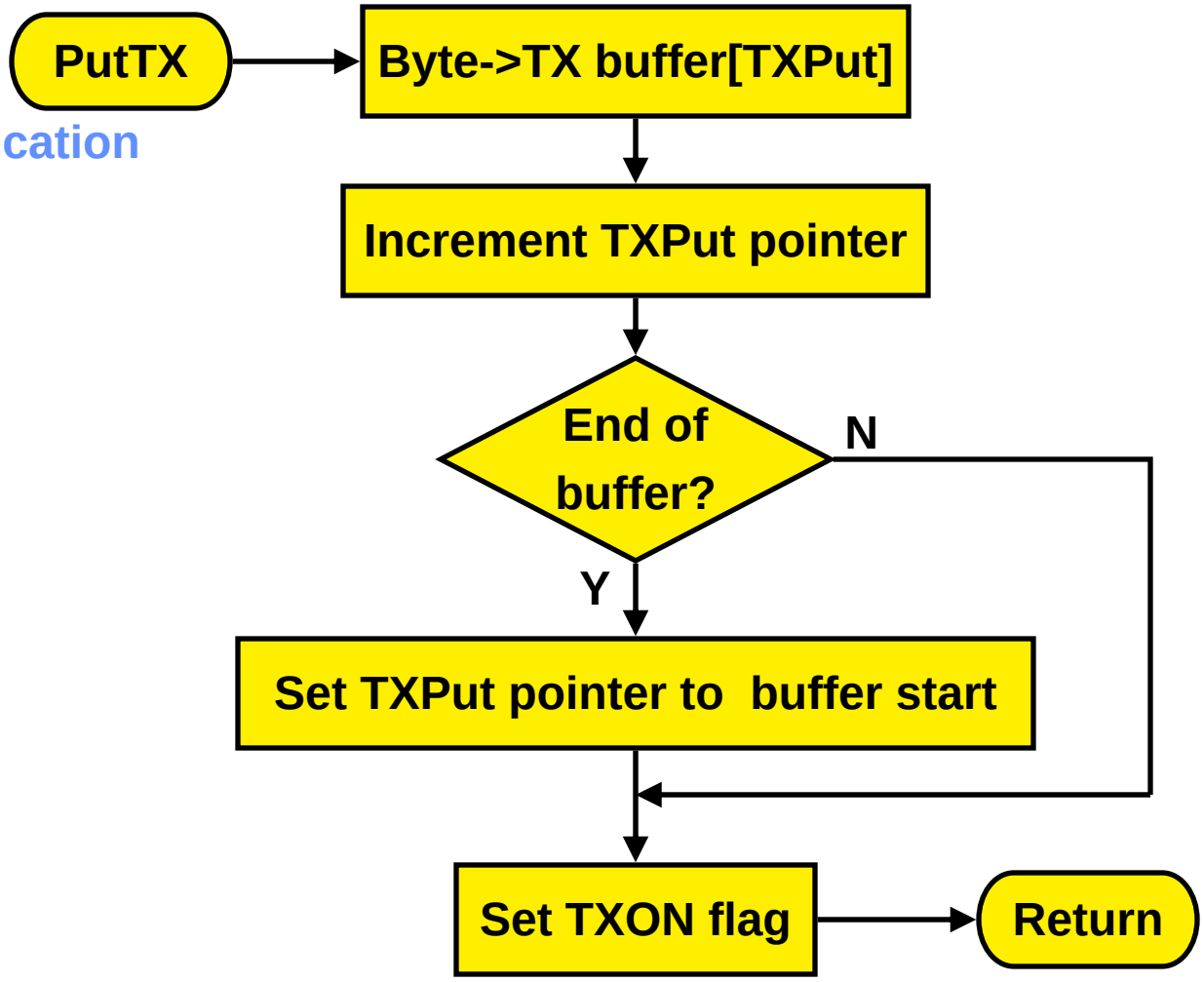


Transmit Buffer

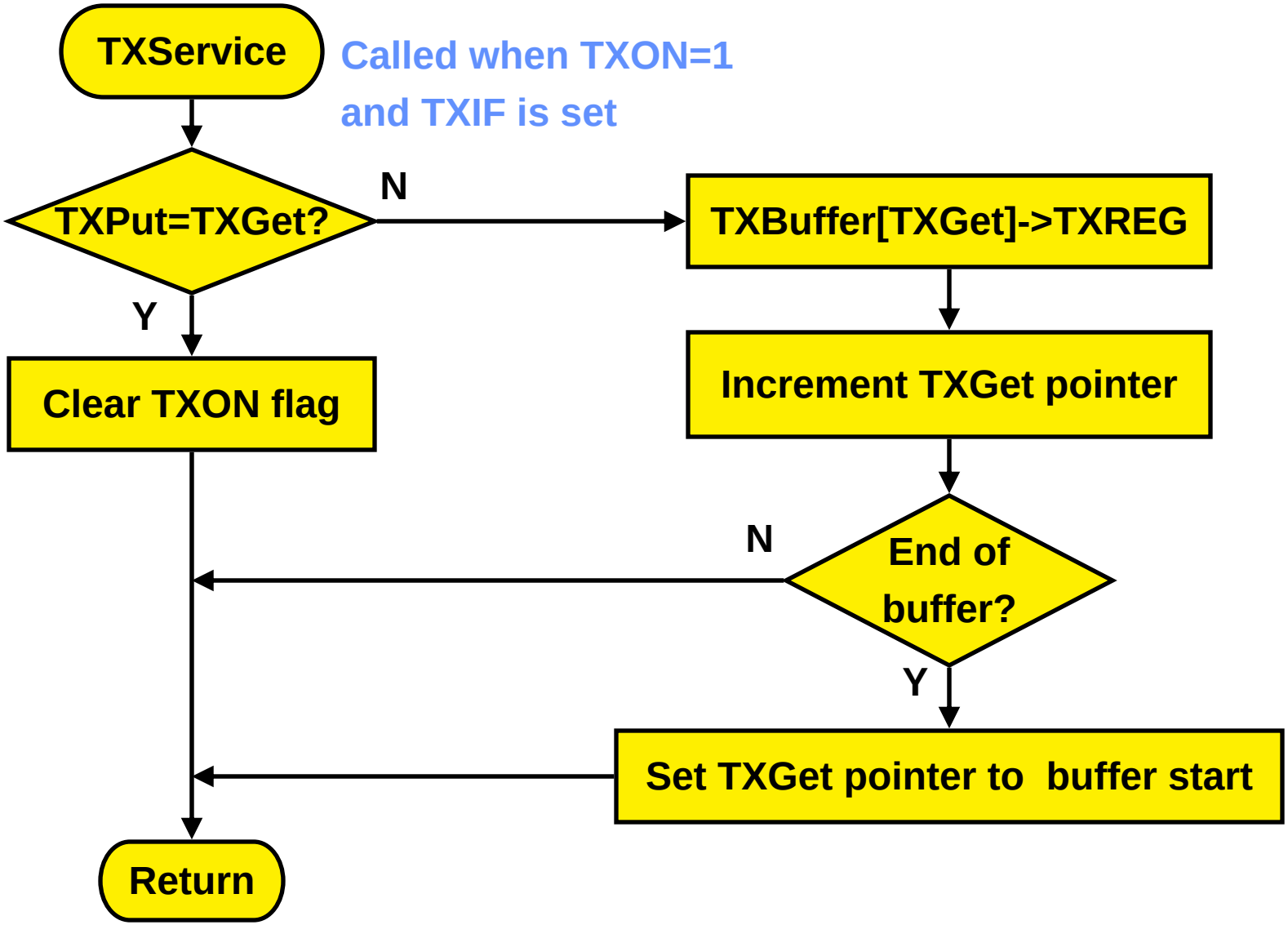
- **As large as longest transmit burst plus margin**
- **Two pointers**
  - TXPut
    - Put pointer for inserting bytes
  - TXGet
    - Get pointer for transferring bytes to TXREG
- **Flags**
  - TXON indicates bytes waiting to be sent
  - TXIF indicates transmit hardware ready

Fill Transmit Buffer

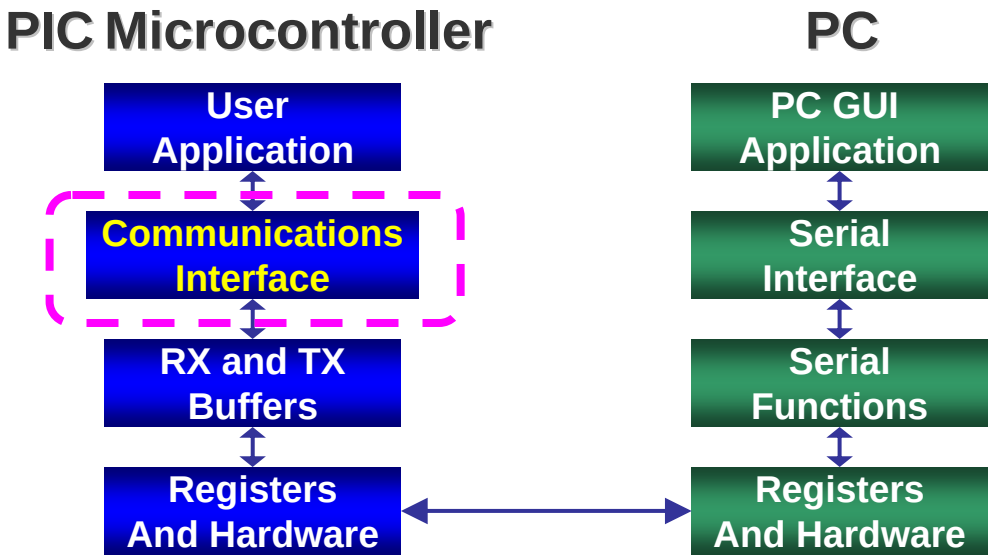
Called at will by application



Empty Transmit Buffer



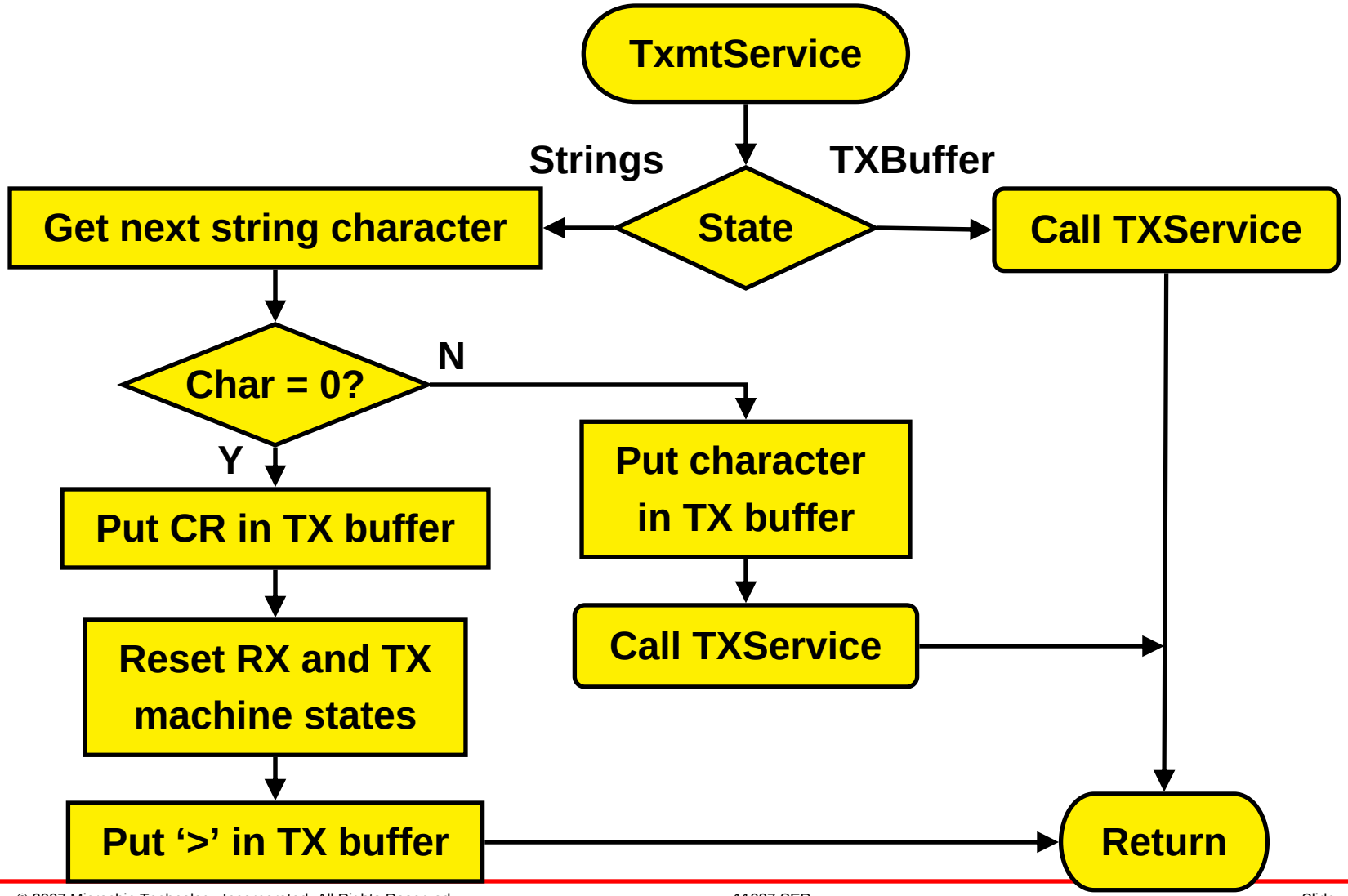
PIC® MCU Firmware Command Interpreter



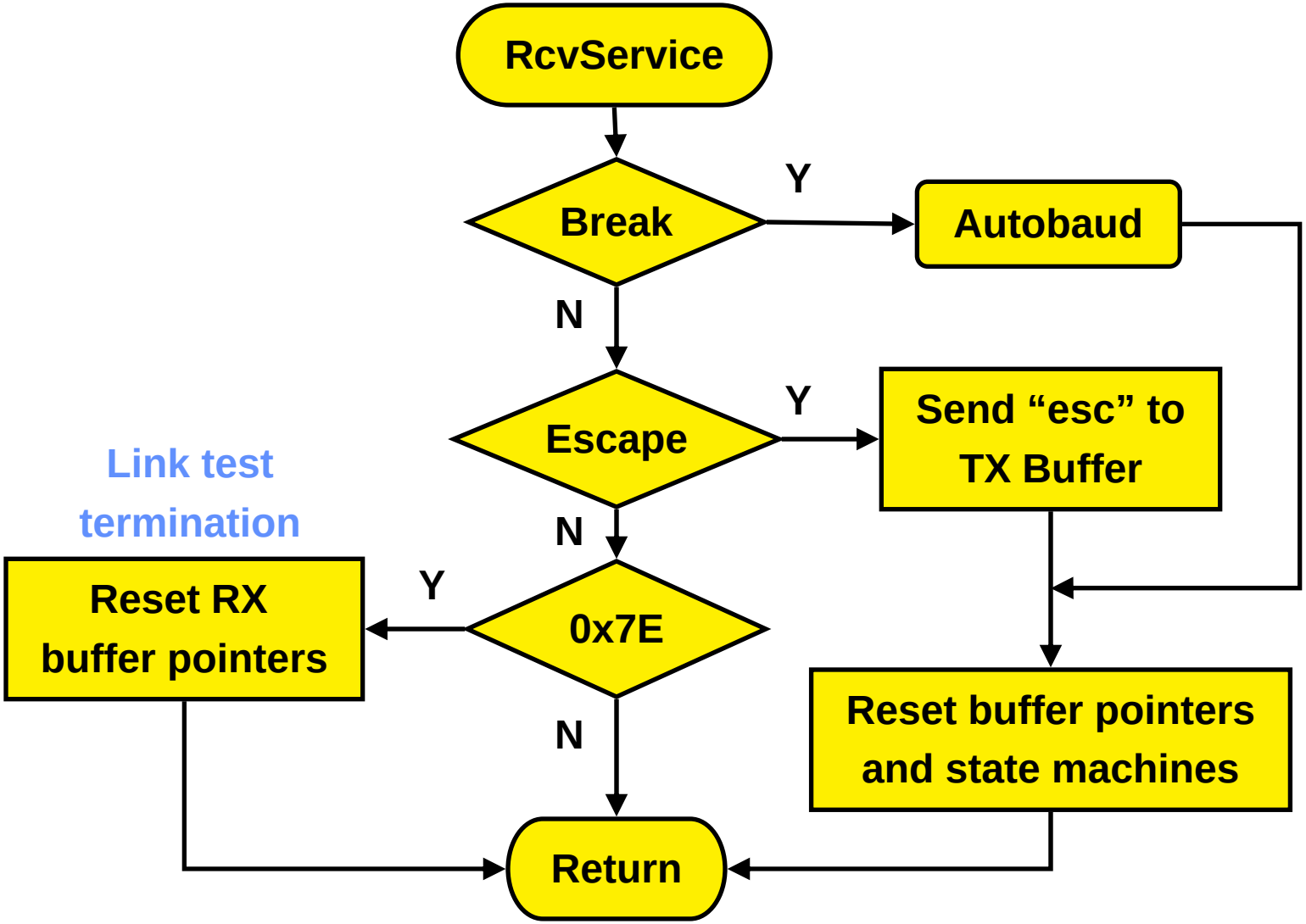
Generic Interface Entry Points

- **MonitorInit**
  - Initialize state machines and call buffer init
- **TxmtService**
  - Transmit strings without overflowing buffer
  - Recognize when TXREG is ready for next char
- **RcvService**
  - Conduit to RXService of buffer level
  - Process BREAK, ESCAPE, and CR characters
- **CmdService:**
  - Decode and process commands in RX buffer

Transmit Service



Receive Service



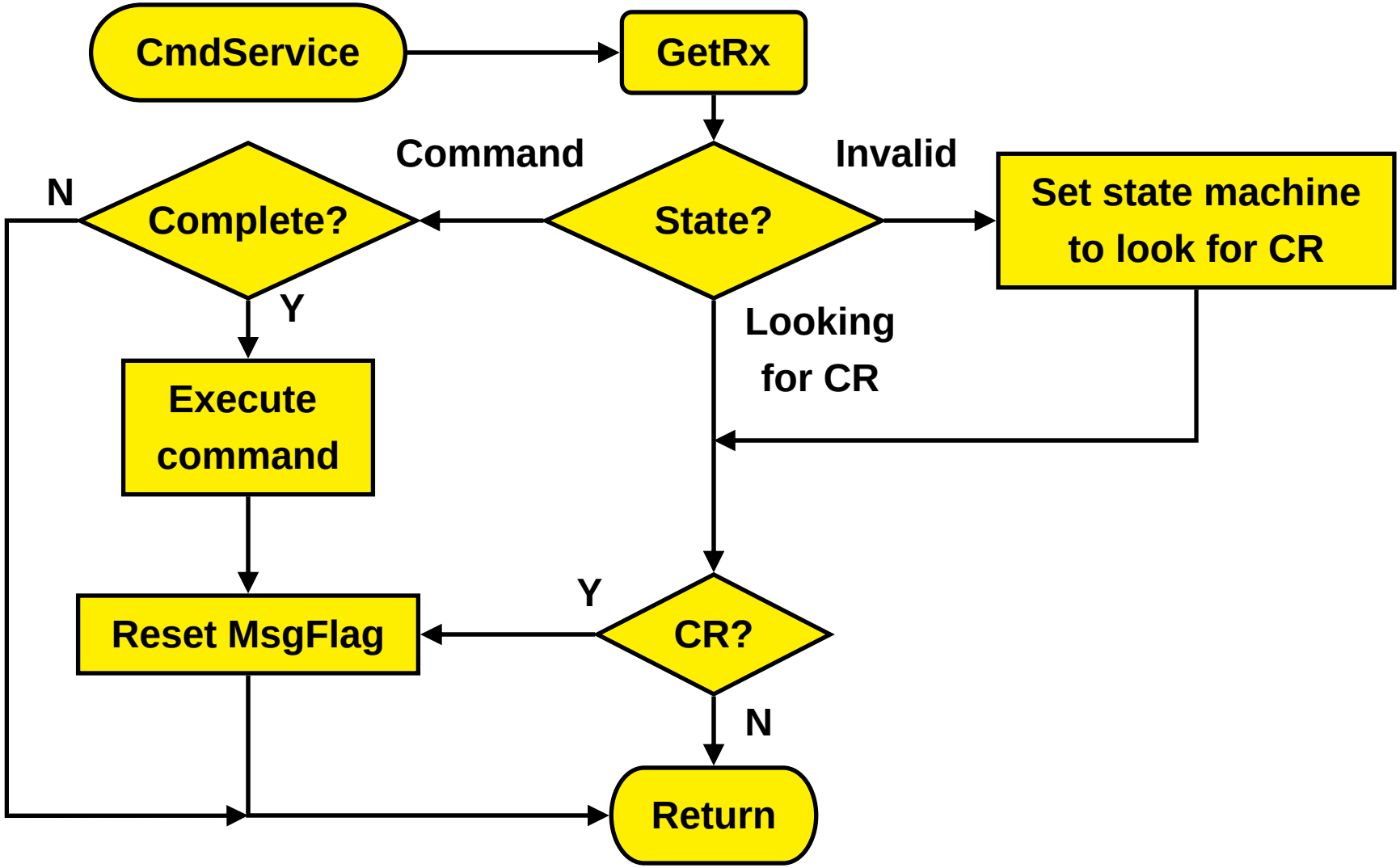
Generic Commands

- Control and monitor application by a few simple generic commands
- Axxx : Display byte at address xxx
- Axxx=yy : Set address xxx to yy
- V : Show firmware version
- ? : Show command menu

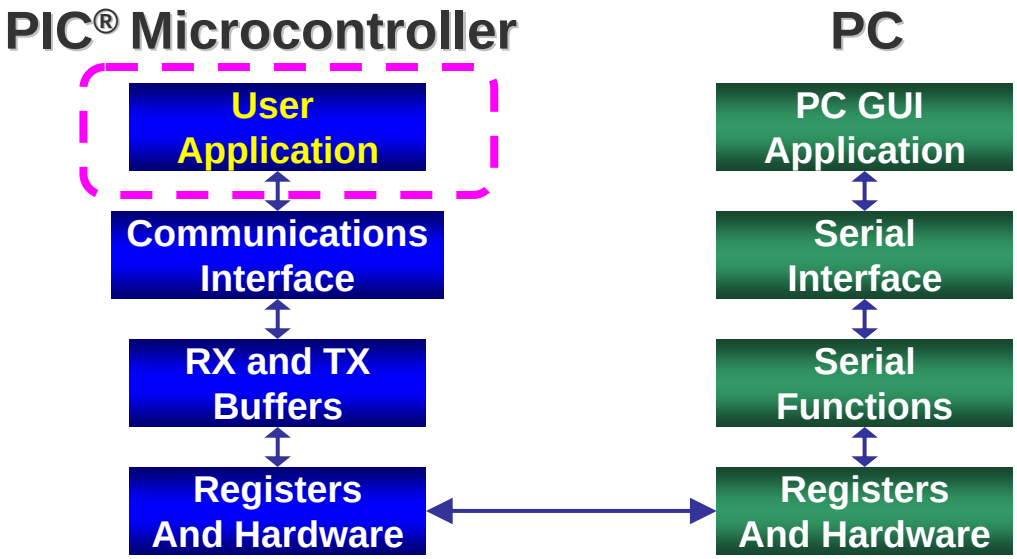
Command Protocol

- Every character received is echoed
- Any invalid character in command invalidates the entire command
- All commands are terminated by CR
- CR is also echoed followed by “>”

Command Service



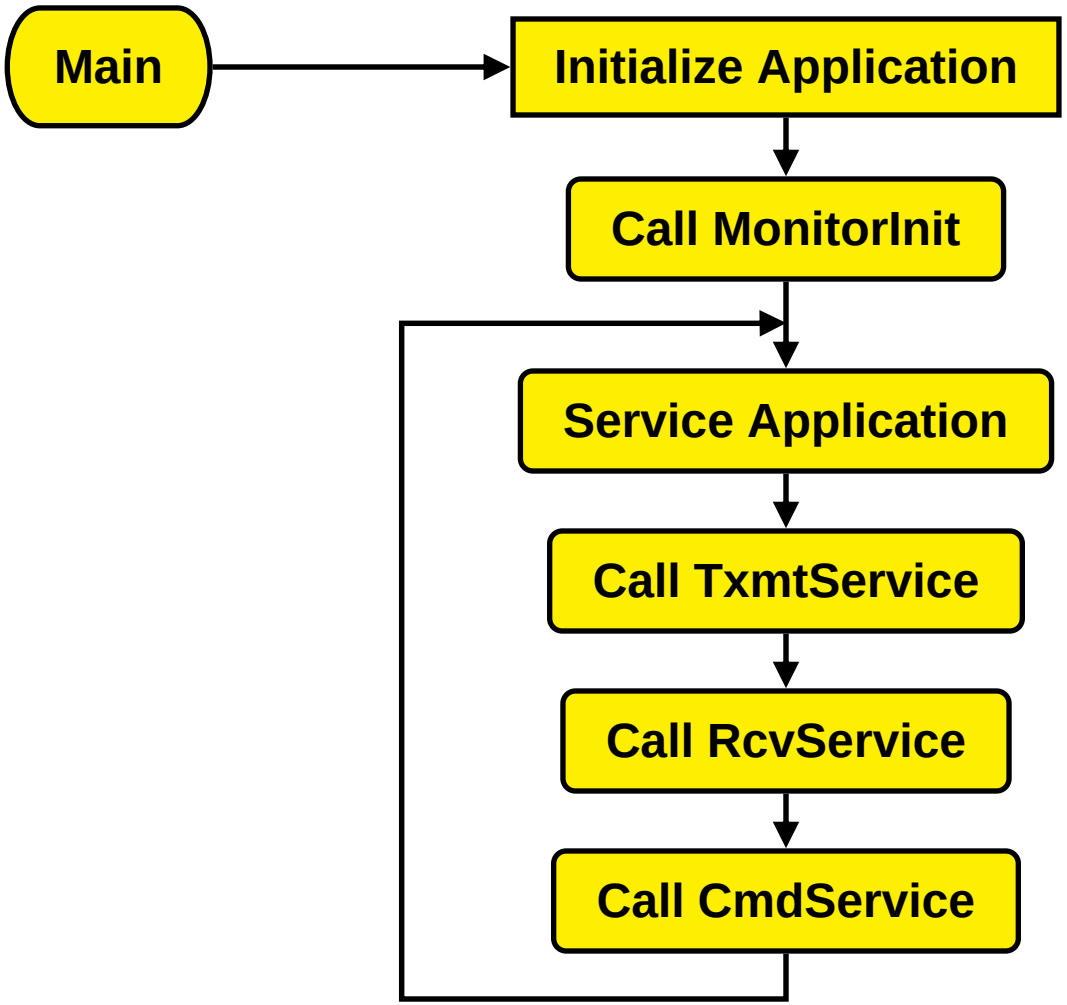
Interfacing User Application to Serial Communications



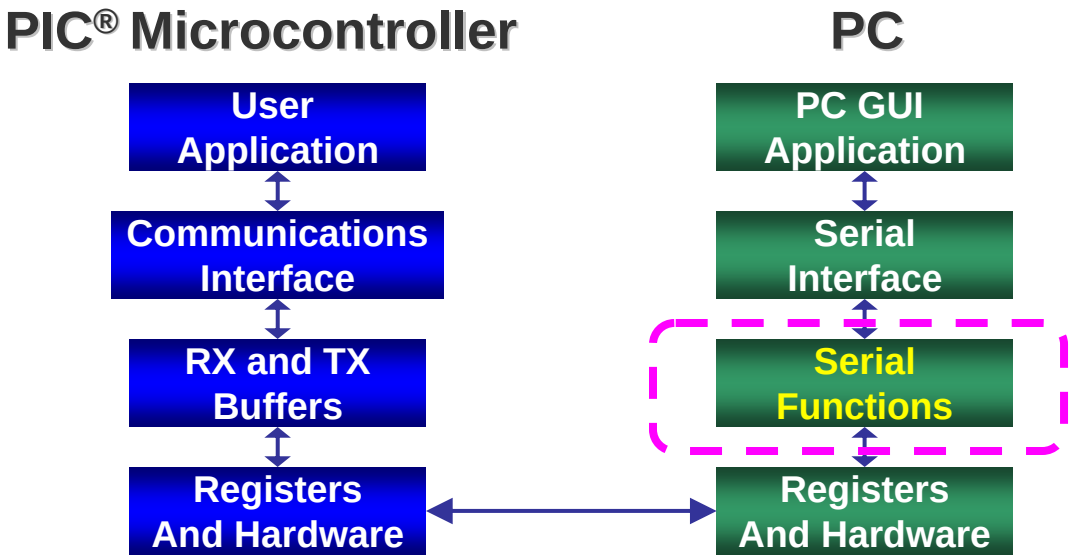
Additions to User Application

- Include call to command interpreter initialization as part of the main application initialization
- Include calls to all three command interpreter services in main application loop
- Average main loop time must be less than one serial character time, otherwise multiple calls to RcvService must be inserted

Example Main Application



Built-In Functions for Serial Communications



Visual Basic (VB) vs C#

- **Both use same .NET serial functions**
- **Same wrappers as C# written in VB**
- **Header references:**
  - C#: using System.IO
  - VB: Imports SYS = System.IO.Ports

System.IO.Ports.SerialPort Methods

- Open()
- Write(buf,ofset,cnt)
- BaseStream.WriteByte(byte)
- ReadByte()
- DiscardInBuffer()
- DiscardOutBuffer()
- Close()
- Dispose()

System.IO.Ports.SerialPort Properties

- BaudRate
- PortName
- BreakState
- ReadBufferSize
- WriteBufferSize
- WriteTimeout
- ReadTimeout
- IsOpen
- Parity
- StopBits
- DataBits
- DtrEnable
- RtsEnable
- Handshake

Finding Available Ports

- **Easy way**
  - VB and C#:

```
MyComboBox.Items.AddRange
  (System.IO.Ports.SerialPort.GetPortNames())
```

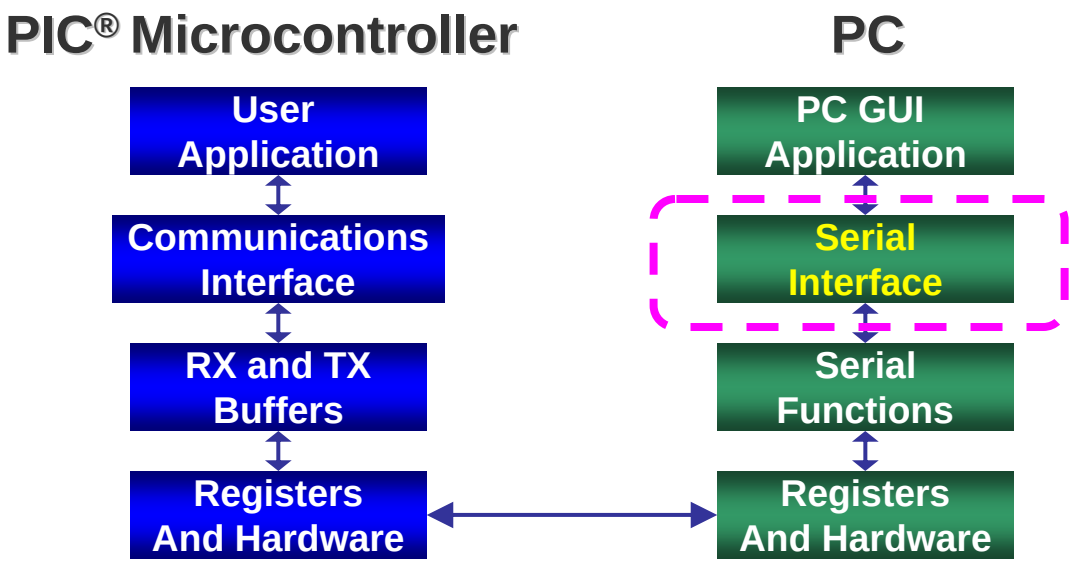
- **Better way**
  - Read registry directly to avoid modems
  - See code listings for example

Opening a Comm Port

- **Allocate memory space by declaring the serial port**
  - VB: Private SPort as SYS.SerialPort

    SPort = new System.IO.Ports.SerialPort()
  - C#: System.IO.Ports.SerialPort SPort = new System.IO.Ports.SerialPort()
- **Initialize the serial port properties**
  - SPort.BaudRate = 9600
- **Open() the port**
  - SPort.Open()

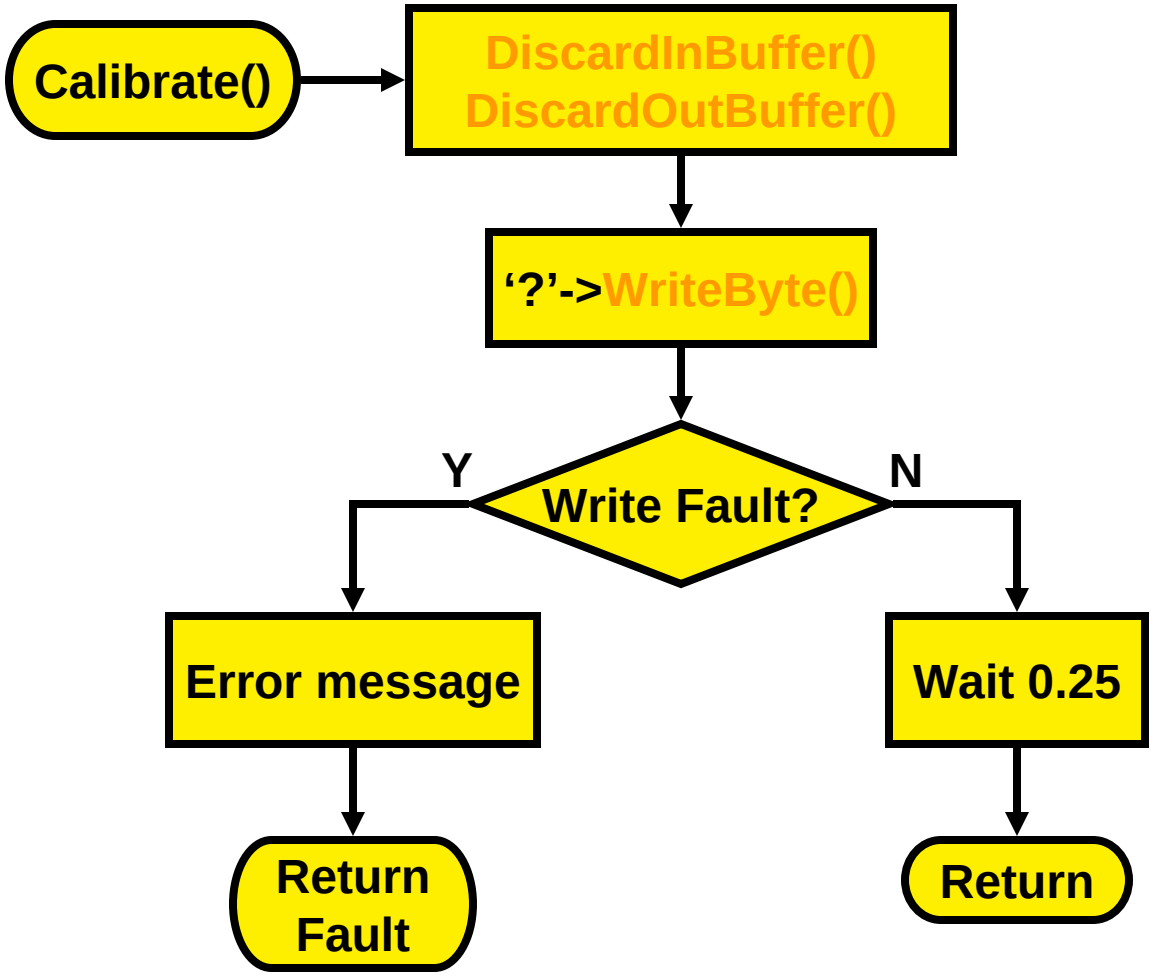
Visual Basic and C# Wrappers For the Serial Port



Serial Interface Functions

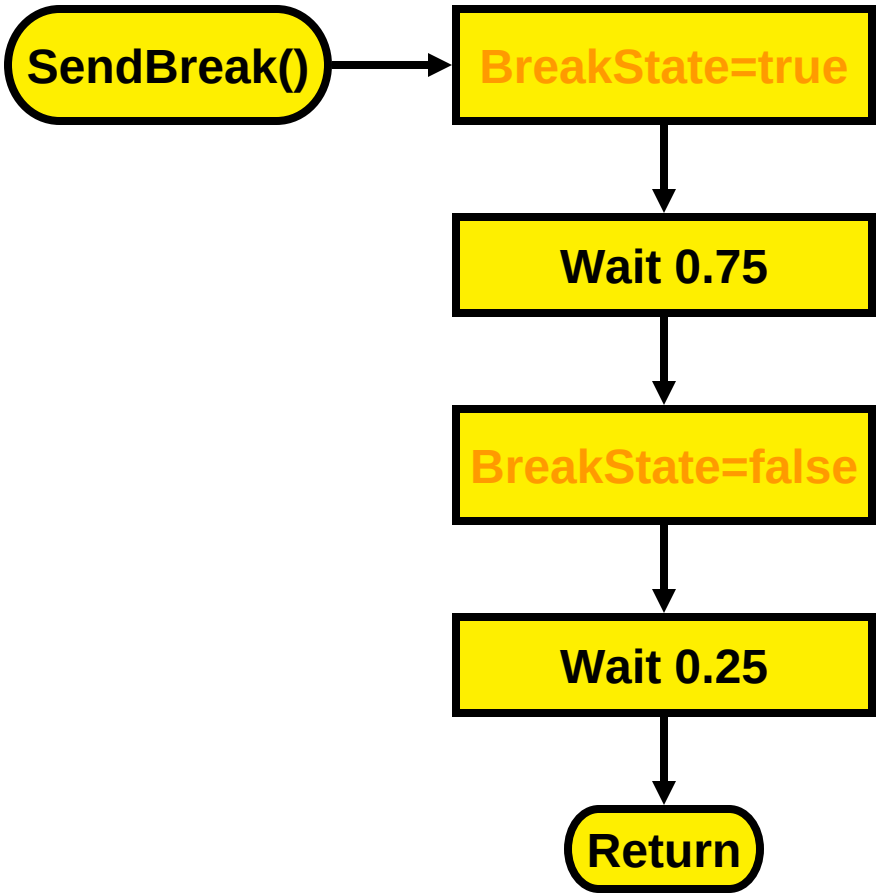
- **Calibrate:** Send calibration character to slave
- **SendBreak:** Request re-calibration
- **Test:** Verify proper communication link
- **SendChar:** Transmit one character
- **ReceiveChar:** Receive one character
- **SendString:** Transmit a string of characters

Calibrate()



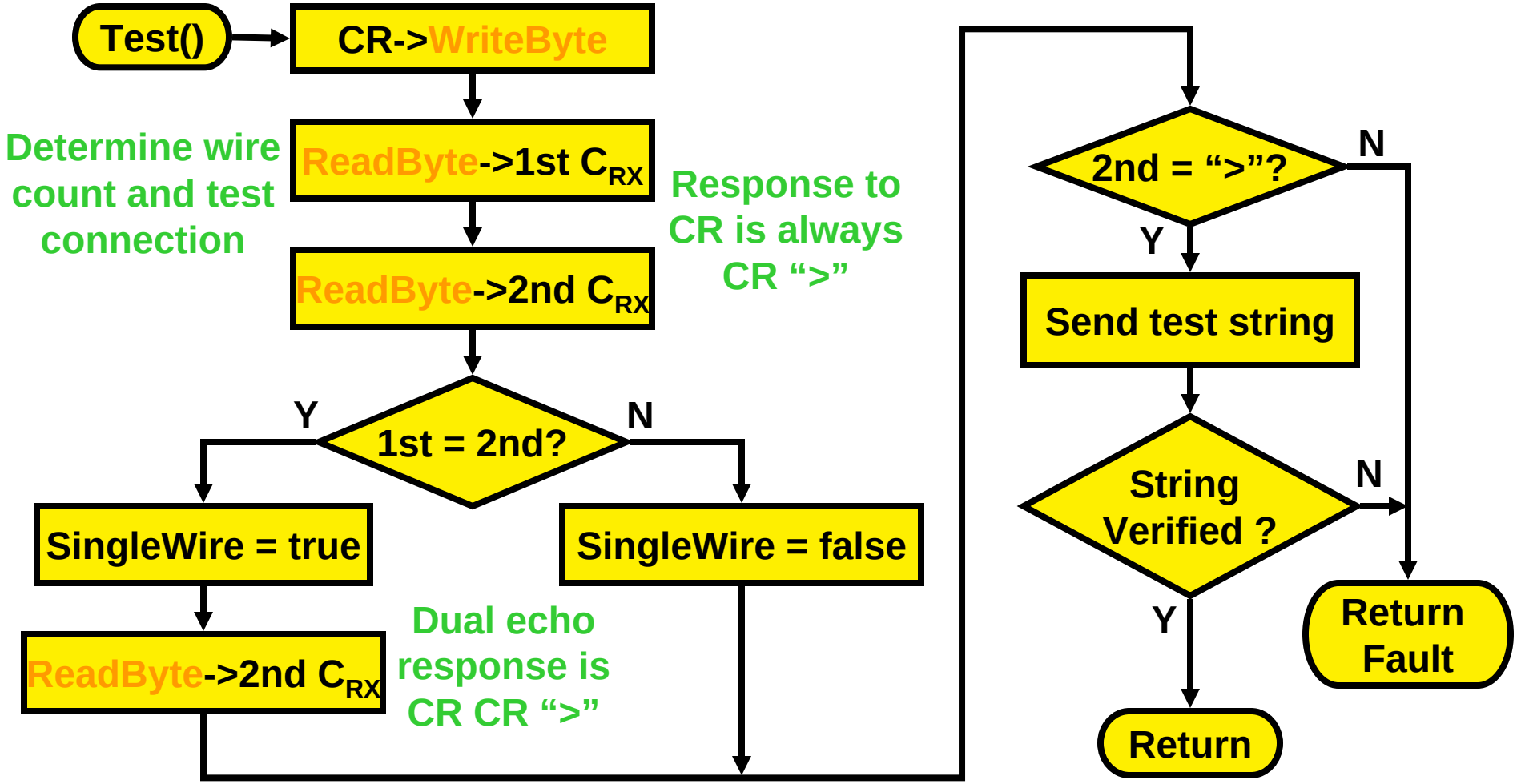
Wait required to allow Autocal to finish and setup for receive

SendBreak()

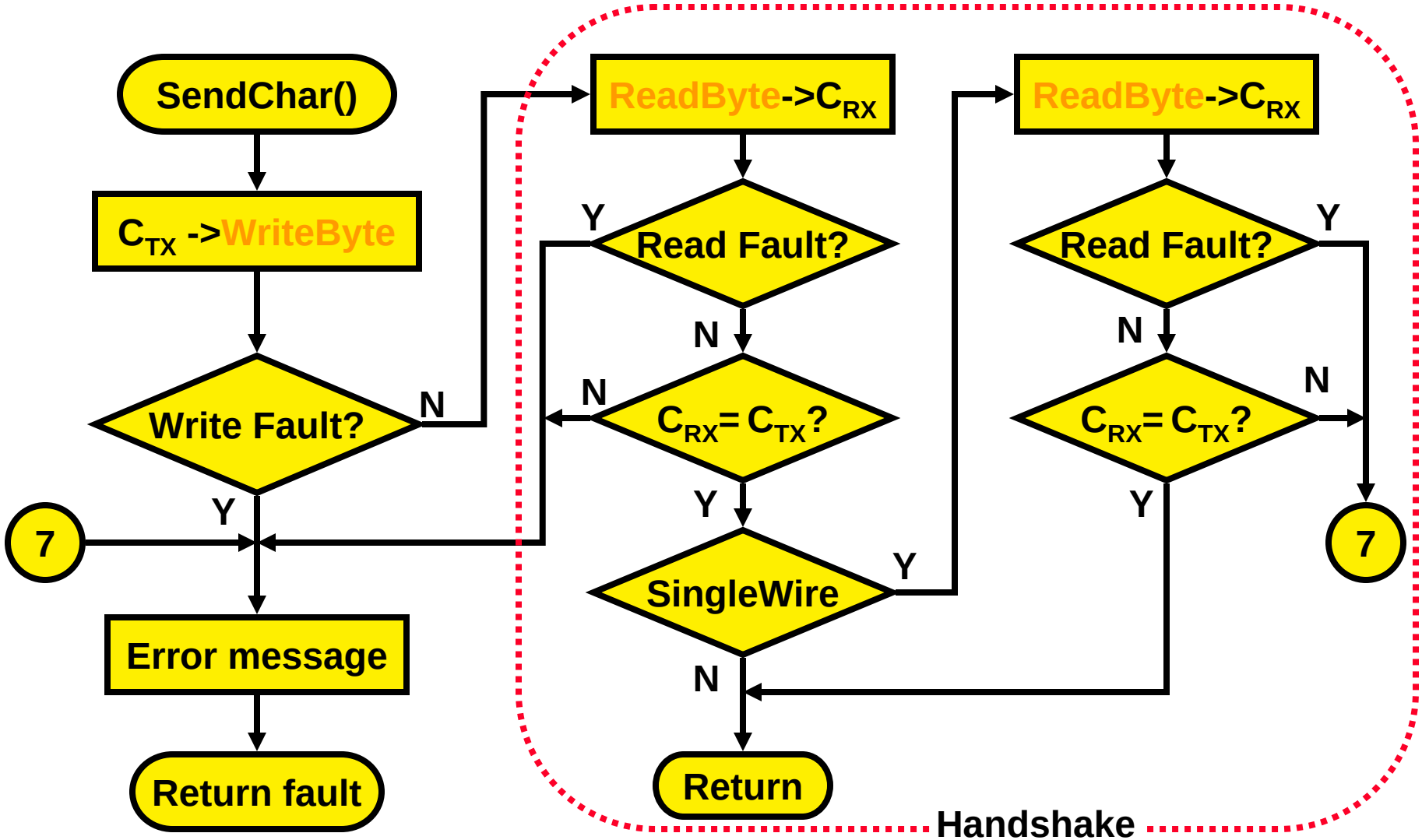


Break time must exceed 1 character time

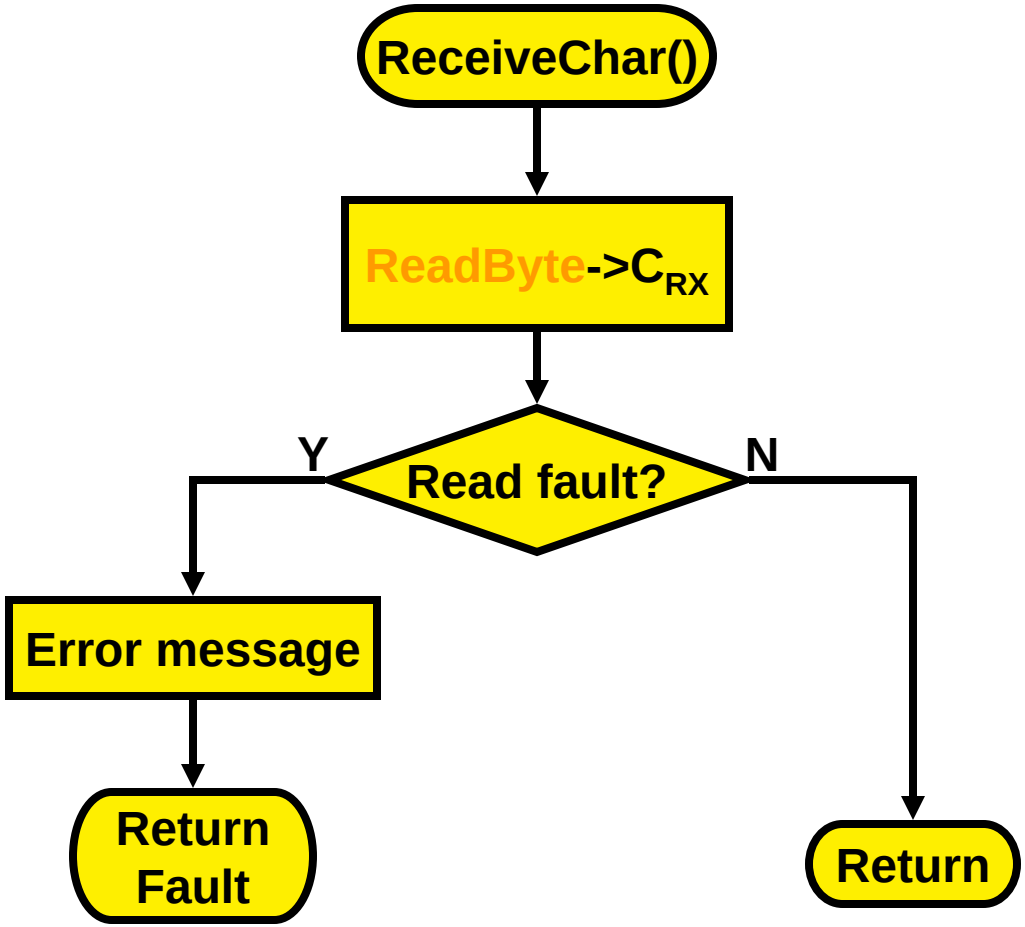
Test()



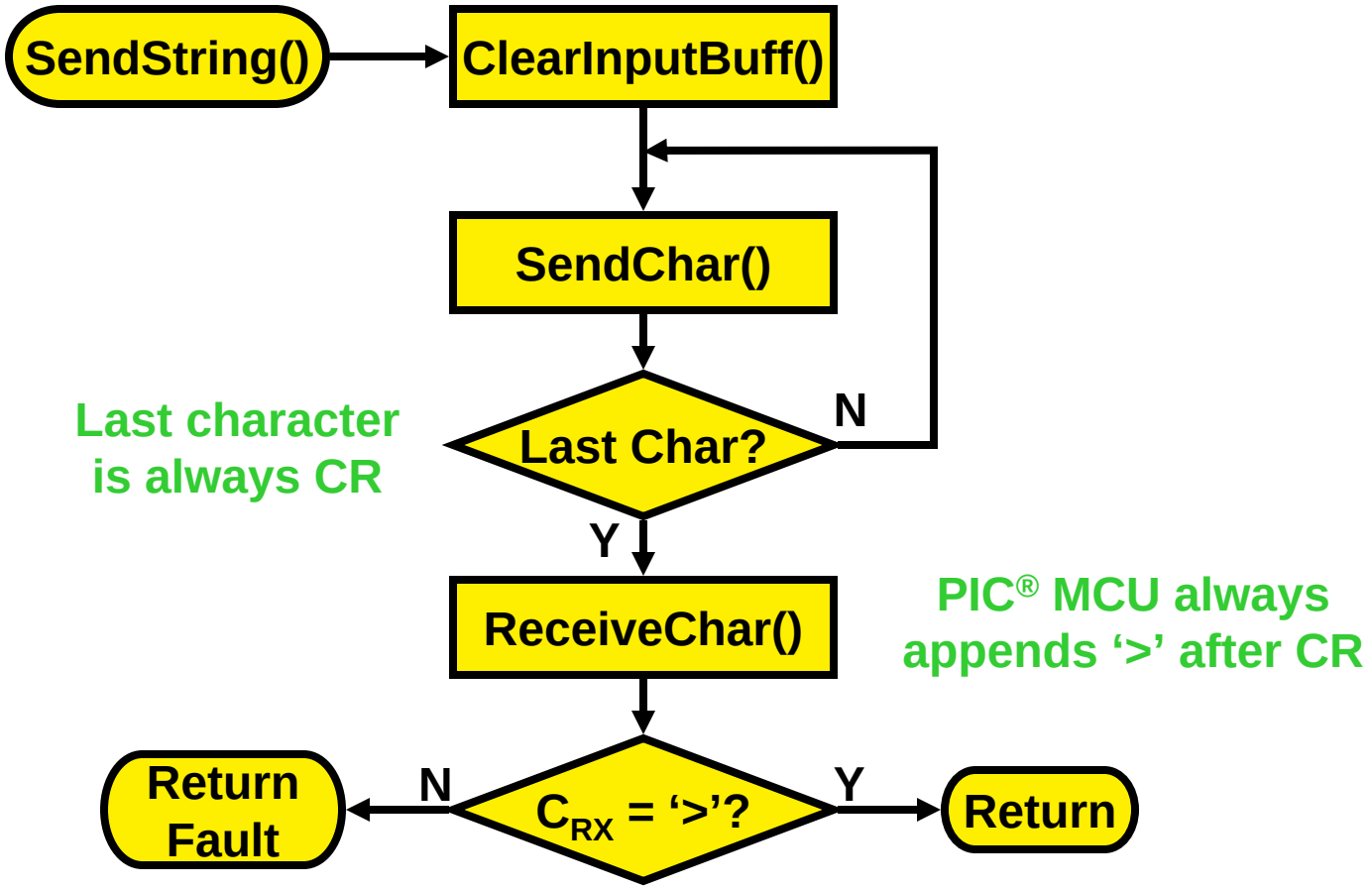
SendChar()



ReceiveChar()

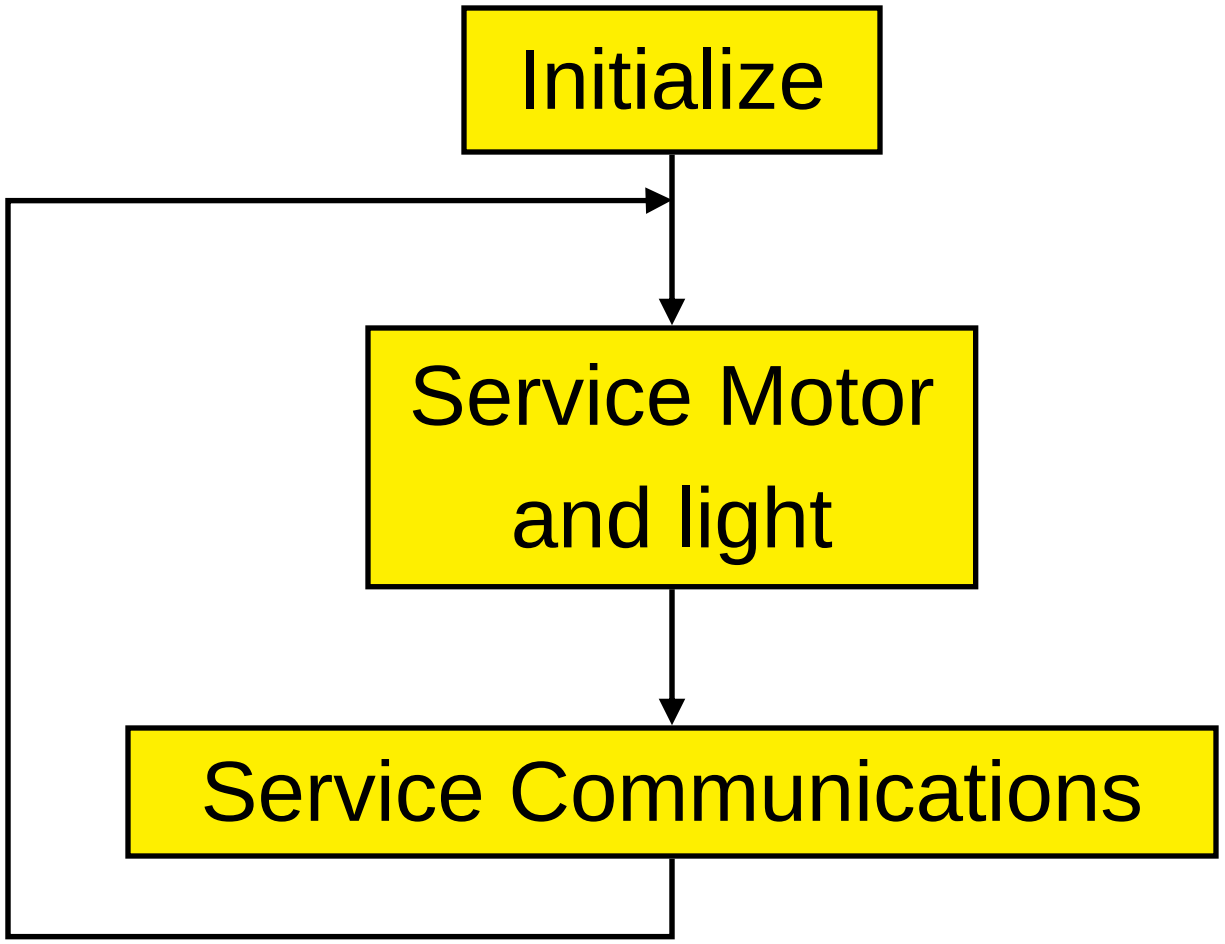


SendString()

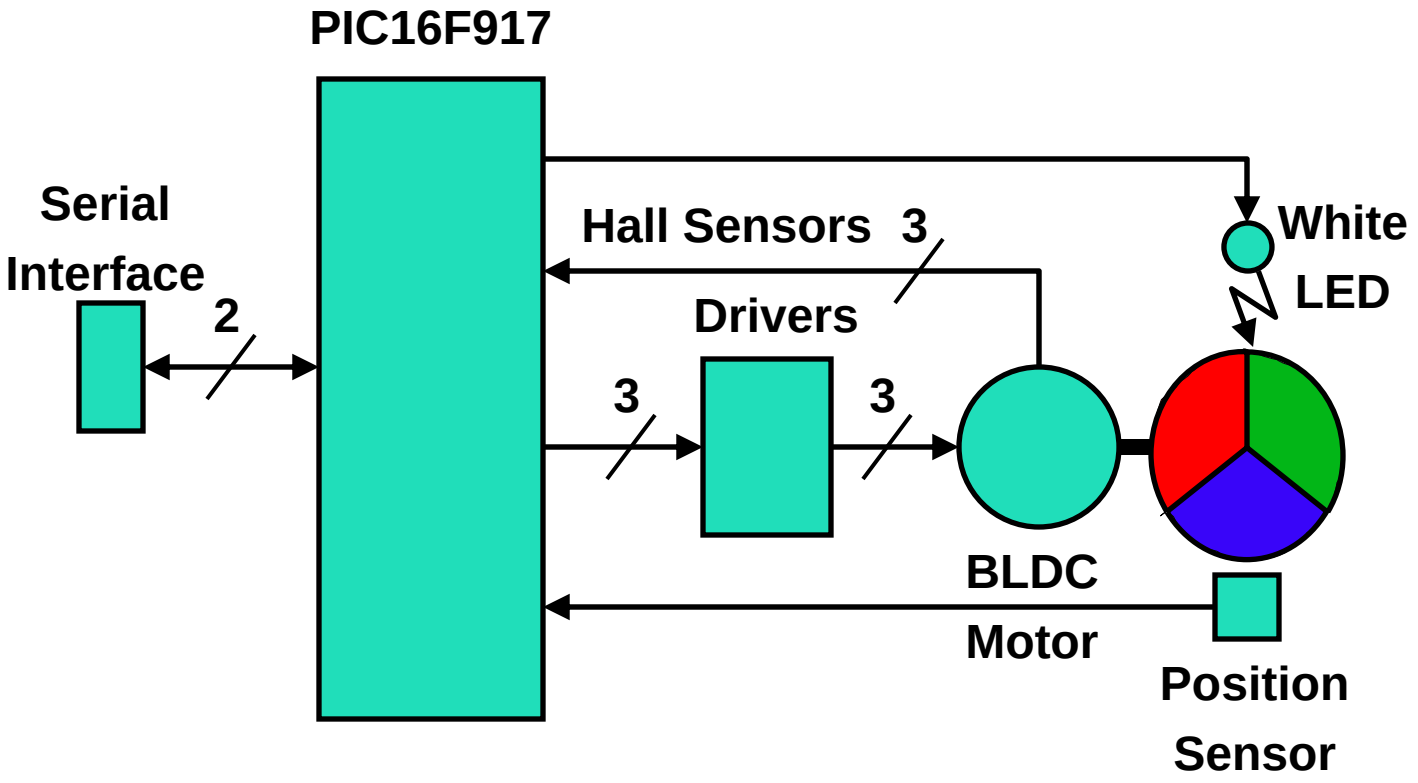


Communication Application Demo

Demo Flowchart



Demo Block Diagram



Summary

- **Elements of communications solution**
  - Hardware: RS-232 to TTL interface
  - Firmware: Polled or interrupt driven UART buffers and command decoder
  - Software: .NET functions for VB and C#
- **Build firmware and software in layers**
- **Read and write SFR and GPR space with minimum universal interface**

Dev Tools used in this class

Color Wheel Demo

- PICDEM™ Mechatronics Demo board (DM163029)
- Standard 9-Pin subminiature D cable
- Hurst brushless motor (DMB0224C10002)
- Keyspan USB to 4 Port Serial adapter (USA-49WLC)

References

- AN712 - Autobaud for the PIC16C5X devices
- AN774 - Asynchronous communications with the PICmicro® USART
- EDN (August 22, 2002) – “Use a PIC® for automatic baud-rate detection”
- DM163029 – PICDEM™ Mechatronics Demo board
- C# For Dummies (ISBN:978-0-7645-9704-3)
- http://msdn.microsoft.com/vstudio/express/

Thanks for Attending

(Please fill out the evaluation form)

UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KeeLoq, KeeLoq logo, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Linear Active Thermistor, Migratable Memory, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.