ELECTRONIC SYSTEMS ENGINEERING **ENEL 351**
FACULTY OF ENGINEERING AND APPLIED SCIENCE
UNIVERSITY OF REGINA

Lab 6

Universal Synchronous/Asynchronous Receiver Transmitter (USART/UART)

Purpose:

This lab will familiarize you with the operation of the Universal Synchronous /Asynchronous Receiver Transmitter (USART/UART) system on the Nucleo-64 Board.

Equipment/Software/References:

- Nucleo-64 Board with USB Cable
- USB to Serial converter cable
- ADALM Unit
- PC with Keil UVision
- STM32F1XX/Nucleo-64 Documentation

Background:

Refer to chapter 23 of the STM32F103xx Reference Manual. The STM32F103RB is equipped with three independent USART systems. Two of these systems are mapped as alternate functions on Port A pins, the other is accessed via Port B. In this lab we will use USART1.

The USART can be configured for either synchronous or asynchronous operation and may be configured to communicate with more complex devices such as smart card or Infrared Data (IRDA) interfaces. We will use it in asynchronous, byte serial mode (UART) to send and receive ASCII data.

The USART contains the circuitry required to convert the parallel data from the microcomputer bus into a serial bit pattern. The USART can transmit either 8 or 9 bits of data within a serial frame and perform parity generation or checking if required.

The registers required for the operation of a USART are:

- USARTx->SR - Status Register
- USARTx->DR - Data Register (read/write)
- USARTx->BRR - Baud Rate Register
- USARTx->CR1 - Control Register 1
- USARTx->CR2 - Control Register 2
- USARTx->CR3 - Control Register 3
- USARTx->GTPR – Guard Time and Prescaler Register

The USARTx_SR register contains the status flags for the USART. The flags can be read periodically by software (polled operation) or can be read following an USART interrupt to determine the interrupt source.

The USARTx_DR register is a read/write register. When read, it is the received data buffer for the USART. When written, it is the transmitter buffer register.

The USARTx_BRR register is used to set transmit and receive baud rates for the USART.

The USARTx_CRx registers are used to set the word length, to enable the transmitter and receiver, and to enable the interrupt system for the USART Only CR1 and CR2 are required to use the USART in UART mode.

The USARTx_GTPR register is used when implementing smart card or IrDA interfaces.

Hardware Setup:

1) Connect the Nucleo-64 Board to your PC using the USB programming cable.

2) Obtain a USB to Serial converter cable from the Lab Instructor or T/A.

a) Connect the USB to Serial converter cable to an available USB port on your PC.

b) Connect the cable’s BLACK “GND” connector to one of the GND connections on the Nucleo board.

c) The cable’s RED “VCC” connector **IS NOT USED , DO NOT CONNECT IT.**

d) Connect the cable’s “TXD” connector to pin PA10, the RxD pin of USART1 on the STM32.

e) Connect the cable’s “RXD” connector to pin PA9, the TxD pin of USART1 on the STM32.

Windows Setup:

1) In the Windows Search Box of your PC, type Device Manager and click on the Device Manager icon that appears.

2) In the Device Manager window, scroll to Ports (COM & LPT). Expand the category.

3) Note the name of the COM port assigned to the USB Serial Port. (COMx)

4) In the Windows Search Box of your PC, type Putty. Click on the puTTy Desktop App icon. (Note: PuTTY may be served by AppsAnywhere instead of being resident on the PC)

5) Click the Serial option under Connection Type. Ensure the speed is 9600 and enter the name of the COM port from step 3 in the Serial Line box. Click Open.

Procedure:

1) Create an initialization routine which will configure USART1 of your STM32F103 to operate at 9600 bps with 8 data bits, 1 stop bit, and no parity.

2) Create a program which will output the ASCII characters beginning with ! and ending with ~ (Hex values $21 through $7E) from the USART1 TX output. The output will be a continuous loop with a new character sent once every second.

3) Test and debug this functionality.

4) Modify the program so that it also monitors the RX input of USART1 to determine when a character has been received. Choose two characters to control the ON/OFF state of the LED connected to PA5.

5) Demonstrate your code to the lab instructor and be prepared to recompile it to use different ASCII characters for part 4 as selected by the lab instructor.

Additional Information: Interfacing the USART to other devices

Devices using 3.3 and 5 volt logic level interfaces

The STM32F103 uses 3.3 volt logic levels on its interface pins. Some of the input pins are capable of accepting 5 volt inputs but all the outputs are limited to 3.3 volts as the maximum they provide. If we are connecting our STM32F103 USART pins to a circuit which is operating at 3.3 volt logic levels, we should be able to use USART1, USART2, or USART3 without any issues.

If the circuit we are connecting to is operating with 5 volt logic levels, we can use USART1 or USART3 only since they use 5 volt tolerant input pins.

Devices using RS232

If you need to connect the STM32F103 USART signals to an RS232 device, you will need a level converter such the MAX3232. The logic level side of this chip, which goes to the USART pins, operates at 3.3 volts. The RS232 side operates at approximately +/- 5 volts to satisfy the RS232 voltage requirements. The beauty of this chip is that it operates from a single 3.3V power supply. You can find breakout boards for MAX3232 devices at Sparkfun.

Devices using USB interfaces

Windows based systems can communicate with your USART via a virtual Com Port and USB/Serial converter. Windows systems will recognize USB/Serial converters based on the FTDI FT232R device and install a Com Port driver for it. Just make sure the USB/Serial converter is using logic level signals on the USART side.