Lab Assignment – Embedded Systems S/W – week 2

Asynchronous Serial Communications

Read this entire document before proceeding with the lab activites.

For oscilloscope screen shots, make sure that you have the InkSaver turned on so that the images are coloured lines on a white background.

Your screen shots must include the date and time as captured by OpenChoice Desktop or saving to a USB drive.

In this lab, you will get some practical experience with asynchronous serial communication and UART programming for the STM32. This lab has 2 parts:

1. Learn to use the asynchronous serial (RS232) decode capabilities of the Tektronix MSO2024B oscilloscope

2. Write a program to configure a UART (asynchronous serial port) on the microcontroller of the STM Discovery Board (***USART1***) and send a message out the serial port. Capture and display that message using the Tek scope.

This is lab will count for 3.75% of your course mark. You may choose to work in pairs and submit a single report for both lab partners; marks will be the same for both partners. You will each be responsible for knowing the complete material in the lab. Working in pairs is not mandatory; you may choose to work individually.

Part 1:

Perform the lab activities with the Tek oscilloscope as outlined in the ***Tektronix UART lab.pdf*** document. Create a Word document containing the following screen shots and associated activities with the shots titled appropriately:

(page references are the pages of the Tek lab document)

- Page 33 + activity – use any character that you happen to capture
- Page 35 – both screen shots
- Page 36
- Page 38 – lower screen shot
- Page 41 – lower screen shot
- Page 44 – lower screen shot

Part 2:

Write a program to perform the task as outlined above. Your program must include the following:

- A **main()** function that calls a function to initialize a UART, and then continuously loops sending a message out the serial port as follows:
  - Your name(s) followed by a line feed character (\n)
  - The text "count: " followed by the value of a 16-bit integer count variable formatted as 4 hexadecimal digits preceded by "0x", followed by a line feed character (\n)

  The main() function should delay for about 100 ms with a delay loop after sending out the message. The count variable printed in the message should increment each time the message is sent.

- A **UARTprintf()** function to be called by main to format and send the message to the serial port.

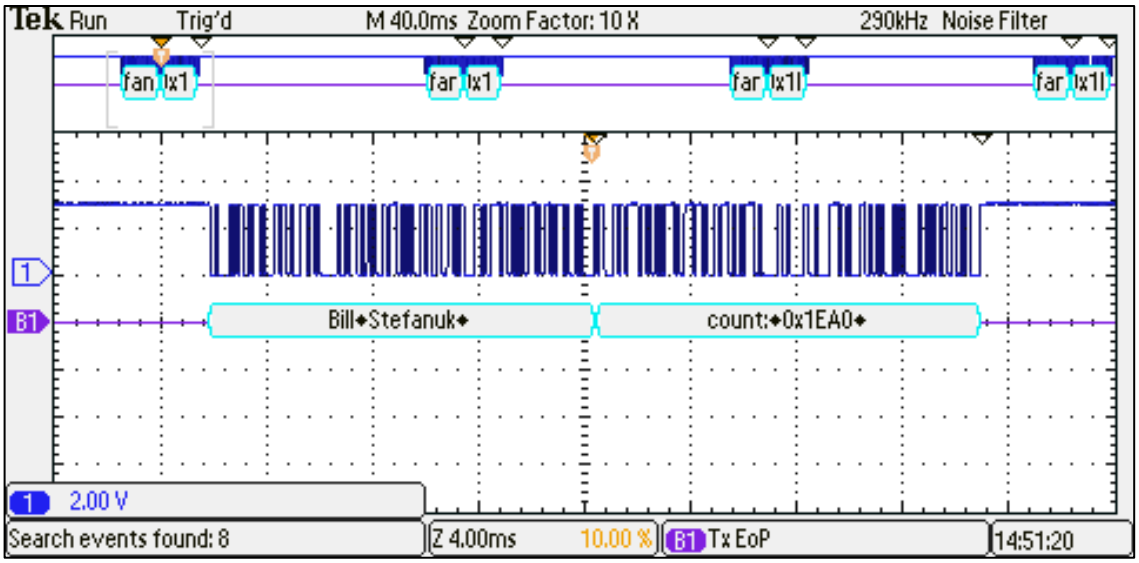
- A **UARTinit()** function to be called by main to configure ***USART1*** as follows:
  - 9600 baud, no parity, 8 data bits, 1 stop bit, no flow control

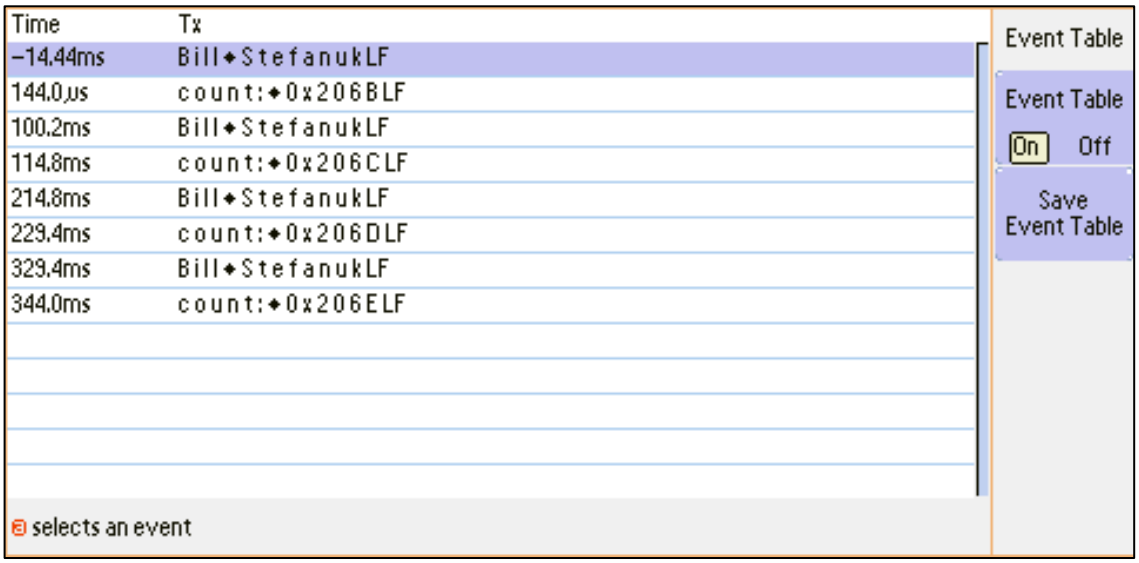
Your code ***must*** be structured as 3 files: **main.c**, **uart.c**, and **uart.h**. You must use USART1 to avoid problems with other devices on the Discovery board.

As always, your program must:

- Use ***Good coding style***
- Include a ***Header block in each file*** giving your name(s), date and the purpose of the program
- Have ***Appropriate Comments*** for each variable, function, and block of code indicating its purpose and general operation. Function comments should include information on parameters and return values
- Use ***White space and Indenting appropriately***
- Have ***No unused (commented out, #ifdef'd out, etc) code***

Demonstrate that your program works correctly by capturing two scope shots as shown below:





Both scope shots should represent the same data. One of the count variable values in the event table should be the same as the value shown on the packet decode screen shot. *(I didn't do that in my example.)* To do this, press the Run/Stop button on the scope so that the scope stops triggering and the display shows **Stop** in the upper left corner before taking both screen shots.

NB: Due to a bug in the oscilloscope firmware, you must turn off the scope channel display (i.e. the normal oscilloscope trace) but leave the Bus display active before capturing the above event table screen shot. Otherwise the scope trace will be superimposed on the event table.

Due: Wednesday 29 Jan, 2020 by 11:59 pm

Submit the following to the Lab 2 eConestoga dropbox:

1. Your Tektronix lab results document following the outline provided in part 1.
2. The source files main.c, uart.c, and uart.h which you wrote to complete part 2 of the lab.
3. A single document containing the two requested oscilloscope screen shots showing the output from your program of part 2.

The 2 parts of the lab have equal mark weighting.