**Lab 2: Universal Asynchronous Receiver Transmitter (UART)**

**Instructor: Prof. Yifeng Zhu**

**Fall 2019**

## Goals

1. Understand the concept of universal asynchronous receiver-transmitter (UART)
2. Learn how to configure, and transmit using UART.
3. Use UART to transmit a message

## Pre-Lab Assignment

1. Complete the pin and timer configuration tables

## Lab Demo

1. Implement myprint(), and UART1\_init().

## Post-Lab Assignment

1. Read Chapter 22.1.5 UART of Textbook
2. Complete the post lab report and write your answer in *readme.md*

In this lab, you will program the STEVAL-FCU001V1 discovery kit to make it communicate with a laptop via UART USB device.

|  |
| --- |
|  |



Figure . Basic UART connection

UART communication use 2 wires: TX and RX. With full-duplex communication, data is always transmitted out bit by bit from the TX line and is received by the other device on its RX line. The receiver reassembles bits received into bytes. In this lab, we will use an USB to TTL Serial Cable, as shown below, to connect the drone control kit and a computer.



Figure . USB to TTL Serial Cable

**Pre-Lab Assignment**

**Lab 13: Interfacing a Bluetooth module via UART**

Student Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TA: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Time & Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Configure GPIO pins**
   1. Enable the USART1 clock.
   2. Code to configure PA10(USART1\_RX) and PA9 (USART1\_TX) to be:
      1. AF mode
      2. Pull-up
      3. High speed
      4. Using Alternation Function 7 (AF7\_USART1)
2. **Configure UART**
   1. We will be using a UART1 which is a USART\_TypeDef object that is already defined in the header files. You need to initialize UART1 to transmit. The three control registers need to be initialized along with the baud rate register. All of this will be done in the UART1\_init() function. Before you initialize UART needs to be disabled, and re-enabled when done initializing it. Set BRR equal to UART\_BRR\_SAMPLING16(HAL\_RCC\_GetPCLK1Freq(),115200 ); This function allows for easily selecting the clock and baud rate. The equation below shows what it returns.

DIV sampling = PCLK2 frequency\*25/(Baud rate \* 4)

BRR = DIV sampling/100<< 4 +( (DIV sampling - DIV sampling \* 16) +50)/100 &0xFF

The above equations simplify to the desired equation where OVER8 is 0

BRR = CLK/ (16 \* Baud rate)

|  |  |  |
| --- | --- | --- |
| Instance | USART1 | Comment |
| Baud Rate | 11 115200 | Bits per second |
| Word Length | 8 bits | Data width |
| Stop Bits | 1 | To synchronize the receiver with incoming data |
| Parity | None | Tells receiving device if any error in the data bits |
| Mode | TX\_RX | Transmit and receive |
| Hardware Flow Control | UA None | Allows the communication device to properly synchronize |
| Over Sampling | UA 16 | Rate to find the center of receive bits |
| RTS | Disable | Request to send |
| CTS | Disable | Clear to send |
| Smartcard | Disable |  |
| Half-duplex Selection | Disable | One device can talk at a time |
| IrDA | Disable | Infrared controller |
| LIN | Disable | Local interconnect network |
| Clock | Disable |  |

**Configure UART1 Peripheral Registers**

Configure the UART1 with the following parameters:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Offset** | **Register** | **31** | **30** | **29** | **28** | **27** | **26** | **25** | **24** | **23** | **22** | **21** | **20** | **19** | **18** | **17** | **16** | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 0x00 | **USART1\_CR1** | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Over8 | RES | UE | M | WAKE | PCE | PS | PEIE | TXEIE | TCIE | RXNIE | IDKEIE | TE | RE | RWU | SBK |
| Value | Reserved | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0x04 | **USART1\_CR2** | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | LIEN | STOP  [1:0] | | CLKEN | CPOL | CPHA | LBCL | Res | LBDIE | LBDL | RES | ADD[3:0] | | | |
| Value | Reserved | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0x08 | **USART1\_CR3** | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. | Res. |  | ONEBIT | CTSIE | CTSE | RTSE | DMAT | DMAR | SCEN | NACK | HDSEL | IRLP | IREN | EIE |
| Value | Reserved | | | | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |

1. **Transmit using UART**
   1. The data register is used to transmit data. Make a loop that runs once for each character of the data to transmit. Set the data register equal to the 8-bit value of that character. Make a while(1){} loop and call myprint() inside of it, and set a break-point their. Have your function send it “Hello world”.
2. **Transmit** 
   1. Download and install Tera-Term: <bit.ly/2VCJS8s>
   2. Use a UART to USB connector: <amzn.to/2M1U1Mi>
   3. Plug in the USB connector, open Tera-Tern, and configure to be a serial port shown below, and then go to Setup->Serial Port, to set up the serial port settings shown below.

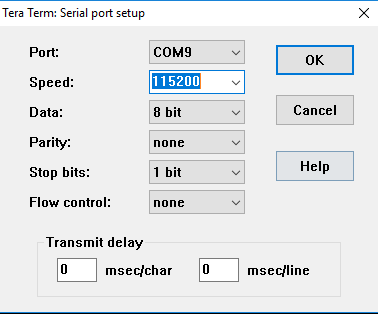
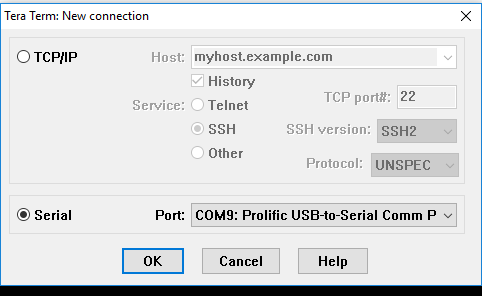
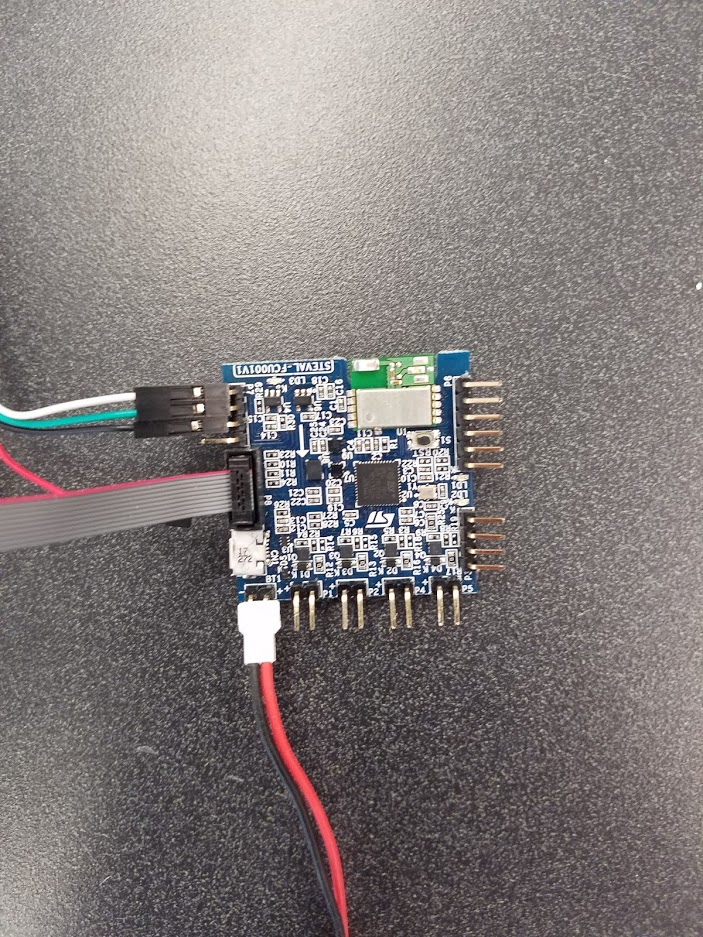


Figure 2: How to configure Tera Term

* 1. Connect the pin closest to the edge of the board that is apart of P7 connector to the Rx (White) pin of the USB connector.



*Figure 3: Board Connections*

* 1. In True Studio flash the board and run the program. You can put a breakpoint at the print statement to make it not constantly print.

**Post-Lab Assignment**

**Lab 2: UART**

Answer the following questions in the file Readme.md and *submit it with your lab code to the gitlab server*.

1. Using an oscilloscope or digital analyzer to capture the signal on the USART\_TX pin of board. What is the bit rate? Is it the same the baud rate? Explain your reason.
2. If the parity bit is used in the data frame, how to detect transmission errors