

**The Hong Kong Polytechnic University**  
**Department of Electronic and Information Engineering**

**EIE3105 Integrated Project (Part I)**

**Laboratory Exercise 4: AVR Serial Port Programming**

**(Deadline: Check the course information)**

**Objective:**

To develop C programs with serial port communication under the Arduino platform.

**Introduction:**

This experiment introduces an application of using the serial port communication inside the AVR microcontroller.

**Equipment:**

Atmel Studio 6  
The Arduino Starter Kit

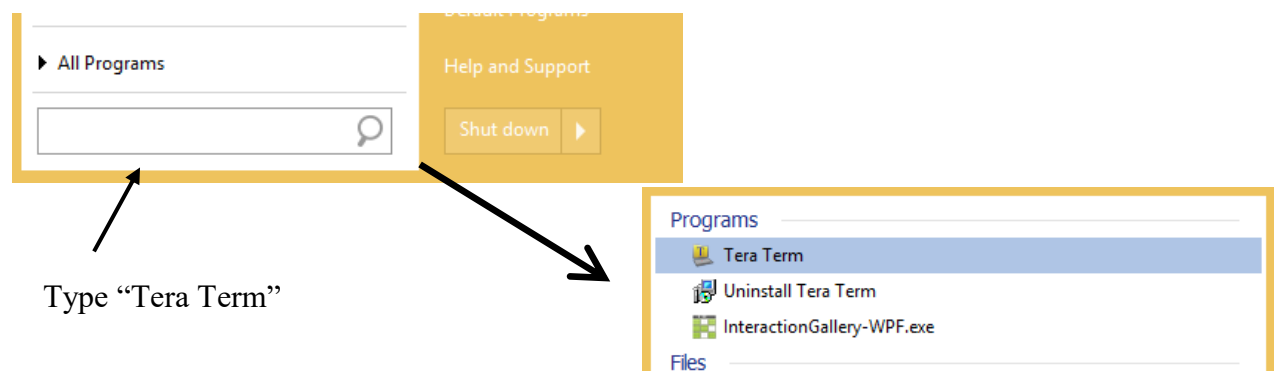
**Procedure:**

*Section A: Write a C program to echo a character*

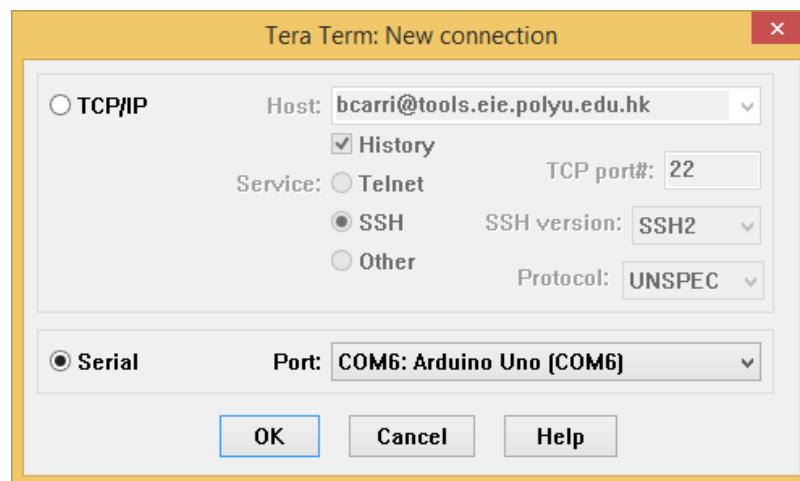
Write a C program to complete the following task:

Receive a character from the serial port by using the polling method and then send it to a PC terminal through the serial port. The baud rate should be 4800.

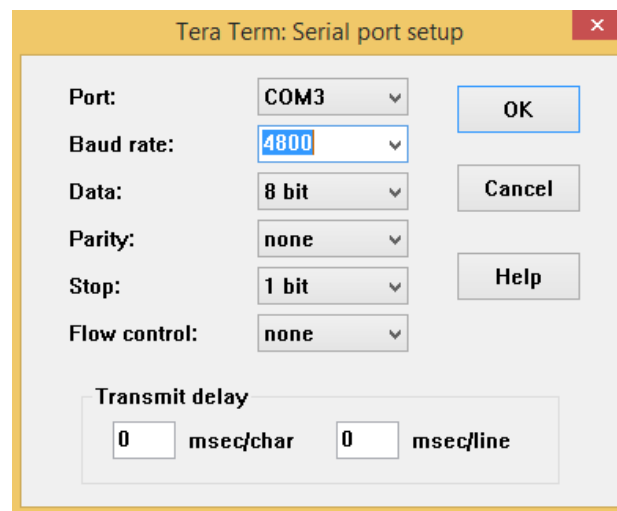
To view the serial port communication, you should use the freeware “Tera Term”, which is a PC terminal. You can use the search function to locate this software in your computer.



Select “Serial” and then an appropriate port for the serial port communication. The port number should be the same as the port that the Arduino connects to your computer.



Select “Setup” from the main menu and then “Serial port”. Set the baud rate to 4800.



Note that you must close Tera Term before you burn your program into Arduino; otherwise, Tera Term holds the serial port and you cannot burn your program successfully.

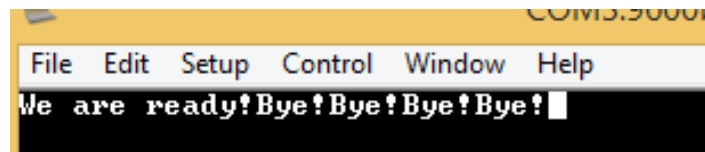
A character can be sent to the Arduino by typing it (i.e., using the keyboard). When you type a character, Tera Term gets the input and sends it to the Arduino through the serial port. When the Arduino gets a character, it will send the character to the serial port and you can read it through Tera Term.

### Section B: Write a C program to send and receive strings

Write a C program to complete the following tasks:

1. (Do it once at the beginning) Send “We Are Ready” to the serial port by using the polling method. Note that the clock frequency of the Arduino Start Kit is 16 MHz.
2. (Do it repeatedly) Receive data from the serial port by using the polling method. If the received string is “Hi”, your program should send “Bye” to the serial port. (Hint: To get a string from the serial port, your program should receive it character by character. For example, if your program receives a string “abc”, the characters received from the serial port should be “a” first, and then “b”, and finally “c”. )

The string “Hi” can be sent to the Arduino by typing it (i.e., using the keyboard). When you type “Hi”, Tera Term gets the input and sends it to the Arduino through the serial port. When the Arduino gets the string, it will send a message “Bye!” to the serial port and you can read it through Tera Term (see the figure below):



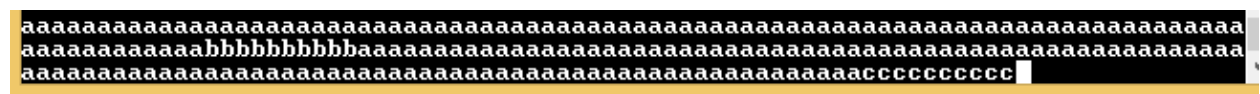
### Section C: Write a C program to keep sending and receiving characters

Write a C program to complete the following tasks using the polling method:

1. Before you press any keys, character ‘a’ is printed continuously.
2. When you press a key (say ‘b’), 10 characters of this key (i.e., ‘b’) are printed out and then stop.
3. After that when you press a key other than the first key (i.e., ‘b’), nothing happens.
4. When you press the key again (i.e., ‘b’), character ‘a’ is printed continuously (i.e., resume).

Set the baud rate of the PC terminal (i.e., Tera Term) to 9600.

If your program runs successfully and the setting of Tera Term is correct, you should see the following output:



### Section D: Write C programs by using the serial port interrupt

Repeat Section B and C but this time you use the serial port interrupt (not the polling method).

**Demonstrate your applications in Section B, C and D to our tutors or technicians.**

**Instructions:**

1. You are required to demonstrate your programs to our tutor or technicians.
2. Zip all programs (including the whole projects) in Section A to D to a single file. Submit it to Blackboard.
3. Deadline: **Check the course information.**

*Ivan Lau  
Lawrence Cheung  
August 2017*