Embedded systems course
UART and Serial port specification

Your device is equipped with a 16550-like UART controller and an RC-232 serial port interface.

For a UART overview you can refer to the "Serial Programming" Wikibook and for detailed register listings please refer to the "8250 UART Programming, UART registers" chapter.

The UART registers are MMIO-based. The base register address is **0x100000.**

The UART asserts the CPU on interrupt line 4.

Your device does not support some of the standard capabilities of a 16550 UART, making MCR, MSR & FCR registers redundant - Only the "Transmitted Data", "Received Data" and "Common Ground" pins are wired (pins 2,3,5), The functionalities controlled by "Modem Control Register" are not implemented and FIFO is not implemented.

The UART connection parameters should be fixed to the following parameters on initialization, as this are the parameters used by the other devices that will communicate with yours:

1. Baud rate = 19200 bps (bits per second)
2. Data bits = 8
3. Parity = None
4. Stop bits = 1
5. Flow control = None

Communicating with your device

In order to test your device you will need a way to connect to its (virtual) serial port and communicate with it.

You are given a C library (embsys_uart_interface.o) and an API (embsys_uart_interface.h) that will allow you to write applications which will run on your host OS and can communicate with your device. Read the documentation inside the API header file to learn how to use it.

'embsys_term' is a simple sample tool based on the API. It reads characters from your shell and after pressing 'Enter' it will send them to your device. Anything the device sends back will be printed to the screen. Enter '!' to exit the program.

TIP: You can use input files with premade sequence of characters in the following way:
`./embsys_term < input_filename |& tail -n 1`