

Linux driver assignments – primer on the serial UART controller and its associated I/O port

1. read class notes related to serial UART controller
2. read data sheet of serial UART controller –
 - a) in our case, section 8.0 will provide detailed information about registers and their usage
 - b) refer to Block diagram 5.0 on page 10 of data-sheet
 - c) refer to register addresses table on page 11 of data-sheet
 - d) refer to Table II on page 14 of data-sheet
3. refer to sample code - `serial_class.c` provided
4. further hints are provided on the next page

Linux driver assignments – primer on serial UART controller and associated I/O port

- a) base address of serial UART controller on your system may be located at 0x3f8 or 0x2f8 or 0x3e8 or 0x2e8(in I/O address space) – based on the particular base address, serial UART controller's register addresses are located at well defined offsets from the base address of serial UART controller
- b) you must initialize the serial UART controller before it can be used - various control registers must be initialized appropriately – read class notes, data-sheet and sample code(serial_class.c) for finding different control registers – understand what they do before coding further !!
- c) data-transfer using serial I/O controller involves writing to Tx h/w buffer and reading from Rx h/w buffer – you will find that base address + offset 0 is used to access Tx h/w buffer when writing and Rx h/w buffer when reading
- d) when writing data, you must check whether Tx h/w buffer is empty using appropriate bit in LSR and then, you are allowed to write upto a maximum of 16 bytes(Tx h/w buffer size) – if you need to write more than 16 bytes, you must check LSR again for Tx h/w buffer empty status and write again

Linux driver assignments – primer on serial UART controller and associated I/O port

- e) when reading data, you must check whether Rx h/w buffer is non-empty using appropriate bit in LSR and read a single byte only – if you need to read more bytes, you must check LSR for Rx h/w buffer non-empty status before every read
- f) serial UART controller works in normal mode and loop-back mode – refer to data-sheet for more details on normal and loop-back mode and also how to enable appropriate mode using MCR control register
- g) you must enable h/w buffers for Tx and Rx using FCR – read data sheet for more details on FCR
- h) if you need to enable h/w interrupt signal generation for certain h/w events in serial UART controller, use IER – there are specific bits to enable Tx h/w buffer event(s), Rx h/w buffer event(s) and events for h/w errors
- i) MCR also has a special bit (OUT2) that must be set for allowing h/w interrupt signal from serial UART to reach interrupt controller on the system – this information may not be present in the data-sheet as it is a machine specific information

Linux driver assignments – primer on serial UART controller and associated I/O port

- j) use control LCR to set DLAB bit to enable baud-rate setting via DLL and DLM values using appropriate registers – refer to Table -III of data-sheet on page 15 for more details – read , analyse and do appropriate settings
