Date: 14-07-2017

Subject Name: EOS / Embedded Linux

Note : you need to read relevant documentation, as you attempt the assignment questions – in addition, you need to understand, before you can go, for implementation

Note : you need to read the lecture notes | SRM | DATA-SHEET | TRM | samples | *.txt files | relevant kernel documentation

Note : after reading the above and during the practicals,
if you still have specific doubts, please write to
babu_krishnamurthy@yahoo.com

Lab Assignment 2:

1. **Explain clearly, how additional UARTs of BeagleBone Black board can be enabled, by modifying the board's device tree sources/dts/dtsi - you need to understand the UARTs available on the board and modify the dts/dtsi, as per device tree rules and hw rules – test, modify, and verify , that the enabled UARTs are visible, in the /proc/device-tree/* of the target kernel – also, verify, that the platform devices of the additional UARTs are visible, under /sys/bus/platform/devices/* and /sys/bus/platform/drivers/ - explore the attributes and details – check, if they are correctly set-up, as required**

2. **Enable I2C1 bus of the BeagleBone blackboard, by modifying the device tree of beaglebone back – modify, test, and verify, using /proc/device-tree/* ,/sys/bus/platform/devices/* and /sys/bus/platform/drivers/* - in addition, based on pickit board pdf, i2c_demo_board.txt and other attached files, modify the device tree of the beaglebone black, for adding support, for i2c client devices – once added, you need test and verify, using /proc/device-tree/* , /sys/bus/i2c/devices/* , and /sys/bus/i2c/drivers/***

3. **Refer to ldd_assignment5's pdfs, in emb_linux_3/ dump provided, during lectures. Refer to files, with file names starting, with char_* - Understand the working of the *.c files – start with, char_pc_drv.c /char_pc_drv.h files – Build and test them on the target system of BeagleBone black – you need to use user-space files of char_pseudo_test.c / char_pseudo_test1.c – refer to pseudo_class.c, for more comments on initializing a char device/driver**

4. Using the samples , from 3) and led_ctrl.c , you need to export a set of char device , for accessing gpio pins , as per project requirements – you can mainly, use ioctl commands / ioctl() system call API and ioctl method of char driver to provide access to a set of gpio pins, as per project requirements

---