**ESE-3014 EMBEDDED SYSTEMS COMMUNICATION PROTOCOLS AND SECURITY**

**LAB 4 Report**

**GROUP No. 2**

|  |  |
| --- | --- |
| **Group Members** | **Student ID** |
| Christy Rachel Philip | C0765535 |
| Fahad Rahman | C0769871 |
| James M Chacko | C0777192 |
| Premil Presannan | C0777191 |
| Vy Nguyen | C0776242 |

**Task 1. Test your UART on BBB by loopback TXD and RXD as our slides.**

**Step 1 checking the available tty on the device**

|  |
| --- |
| debian@beaglebone:~$ ls -l /dev/ttyO\* lrwxrwxrwx 1 root root 5 Mar 10 06:32 /dev/ttyO0 -> ttyS0 lrwxrwxrwx 1 root root 5 Mar 10 06:32 /dev/ttyO1 -> ttyS1 lrwxrwxrwx 1 root root 5 Mar 10 06:32 /dev/ttyO2 -> ttyS2 lrwxrwxrwx 1 root root 5 Mar 10 06:32 /dev/ttyO3 -> ttyS3 lrwxrwxrwx 1 root root 5 Mar 10 06:32 /dev/ttyO4 -> ttyS4 lrwxrwxrwx 1 root root 5 Mar 10 06:32 /dev/ttyO5 -> ttyS5 |

**Step 2: Adding the command line to enable the ttyO4 in uEnv.txt file**

|  |
| --- |
| **uboot\_overlay\_addr4=/lib/firmware/BB-UART4-00A0.dtbo** |

**uEnv.txt file:**

|  |
| --- |
| **#Docs: http://elinux.org/Beagleboard:U-boot\_partitioning\_layout\_2.0 uname\_r=4.14.108-ti-r131 #uuid= #dtb= ###U-Boot Overlays### ###Documentation: http://elinux.org/Beagleboard:BeagleBoneBlack\_Debian#U-Boot\_Overlays ###Master Enable enable\_uboot\_overlays=1 ### ###Overide capes with eeprom #uboot\_overlay\_addr0=/lib/firmware/<file0>.dtbo #uboot\_overlay\_addr1=/lib/firmware/<file1>.dtbo #uboot\_overlay\_addr2=/lib/firmware/<file2>.dtbo #uboot\_overlay\_addr3=/lib/firmware/<file3>.dtbo ### ###Additional custom capes uboot\_overlay\_addr4=/lib/firmware/BB-UART4-00A0.dtbo #uboot\_overlay\_addr5=/lib/firmware/<file5>.dtbo #uboot\_overlay\_addr6=/lib/firmware/<file6>.dtbo #uboot\_overlay\_addr7=/lib/firmware/<file7>.dtbo ### ###Custom Cape #dtb\_overlay=/lib/firmware/<file8>.dtbo ### ###Disable auto loading of virtual capes (emmc/video/wireless/adc) #disable\_uboot\_overlay\_emmc=1 #disable\_uboot\_overlay\_video=1 #disable\_uboot\_overlay\_audio=1 #disable\_uboot\_overlay\_wireless=1 #disable\_uboot\_overlay\_adc=1 ### ###PRUSS OPTIONS ###pru\_rproc (4.14.x-ti kernel) uboot\_overlay\_pru=/lib/firmware/AM335X-PRU-RPROC-4-14-TI-00A0.dtbo ###pru\_rproc (4.19.x-ti kernel) #uboot\_overlay\_pru=/lib/firmware/AM335X-PRU-RPROC-4-19-TI-00A0.dtbo ###pru\_uio (4.14.x-ti, 4.19.x-ti & mainline/bone kernel) #uboot\_overlay\_pru=/lib/firmware/AM335X-PRU-UIO-00A0.dtbo ### ###Cape Universal Enable enable\_uboot\_cape\_universal=1 ### ###Debug: disable uboot autoload of Cape #disable\_uboot\_overlay\_addr0=1 #disable\_uboot\_overlay\_addr1=1 #disable\_uboot\_overlay\_addr2=1 #disable\_uboot\_overlay\_addr3=1 ### ###U-Boot fdt tweaks... (60000 = 384KB) #uboot\_fdt\_buffer=0x60000 ###U-Boot Overlays### ###U-Boot Overlays### cmdline=coherent\_pool=1M net.ifnames=0 lpj=1990656 rng\_core.default\_quality=100 quiet #In the event of edid real failures, uncomment this next line: #cmdline=coherent\_pool=1M net.ifnames=0 lpj=1990656 rng\_core.default\_quality=100 quiet video=HDMI-A-1:1024x768@60e ##enable Generic eMMC Flasher: ##make sure, these tools are installed: dosfstools rsync #cmdline=init=/opt/scripts/tools/eMMC/init-eMMC-flasher-v3.sh** |

**Step 3: Installing minicom:**

|  |
| --- |
| **debian@beaglebone:~$ sudo apt install minicom Reading package lists... Done Building dependency tree  Reading state information... Done The following additional packages will be installed:  lrzsz The following NEW packages will be installed:  lrzsz minicom 0 upgraded, 2 newly installed, 0 to remove and 95 not upgraded. Need to get 344 kB of archives. After this operation, 1,313 kB of additional disk space will be used. Do you want to continue? [Y/n] y Get:1 http://deb.debian.org/debian stretch/main armhf lrzsz armhf 0.12.21-8 [84.1 kB] Get:2 http://deb.debian.org/debian stretch/main armhf minicom armhf 2.7-1.1 [260 kB] Fetched 344 kB in 3s (95.3 kB/s)  Selecting previously unselected package lrzsz. (Reading database ... 39097 files and directories currently installed.) Preparing to unpack .../lrzsz\_0.12.21-8\_armhf.deb ... Unpacking lrzsz (0.12.21-8) ... Selecting previously unselected package minicom. Preparing to unpack .../minicom\_2.7-1.1\_armhf.deb ... Unpacking minicom (2.7-1.1) ... Setting up minicom (2.7-1.1) ... Setting up lrzsz (0.12.21-8) ... Processing triggers for man-db (2.7.6.1-2) ... debian@beaglebone:~$** |

**Step 4: using minicom:**

|  |
| --- |
| debian@beaglebone:~$ minicom -b 115200 -o -D /dev/ttyO4 |

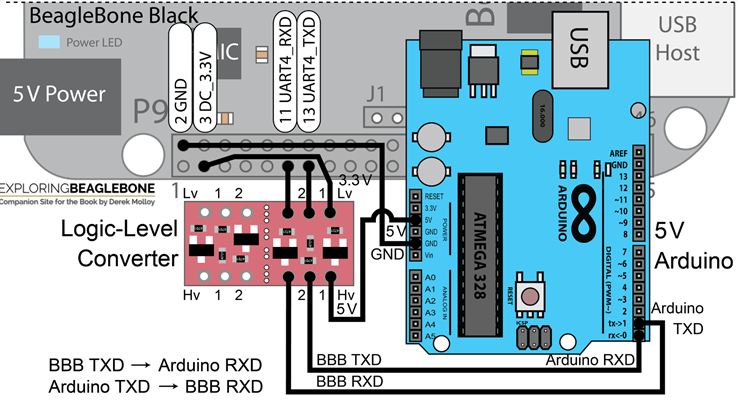
**Step 5: Testing:**

|  |
| --- |
| Welcome to minicom 2.7    OPTIONS: I18n  Compiled on Apr 22 2017, 09:14:19.  Port /dev/ttyO4, 06:32:50    Press CTRL-A Z for help on special keys    aassxxrrhhiihheelllloo hheelllloo |

**Task 2:**

**Setting the UART interfacing between BBB and Arduino:**

Following this connection:



Yes, we do need a logic level control since we using 5V on the Arduino and 3,3V on Beaglebone Black.

**Task 3:** Writing a program to control LED on Arduino board:

Code:

|  |
| --- |
| void setup() {  Serial.begin(115200);  pinMode(13, OUTPUT);   digitalWrite(13, HIGH);  delay(2000);  digitalWrite(13, LOW);  delay(500); } void loop() {  byte CharIn;  digitalWrite(13,LOW);  if(Serial.available())  {  CharIn = Serial.read();  switch (CharIn)  {  case '1':  Serial.write("A");  for(size\_t i =0; i!=10; i++)  {  digitalWrite(13, HIGH);  delay(500);  digitalWrite(13, LOW);  delay(500);  }  break;  case '2':  Serial.write("B");  for(size\_t i =0; i!=10; i++)  {  digitalWrite(13, HIGH);  delay(200);  digitalWrite(13, LOW);  delay(200);  }  break;  case '3':  Serial.write("C");  for(size\_t i =0; i!=10; i++)  {  digitalWrite(13, HIGH);  delay(1000);  digitalWrite(13, LOW);  delay(1000);  }  break;  }  } } |

Testing:

Using minicom from BBB to testing the LED on the Arduino.

There are four cases of the LED

case 1: when the Arduino receives 1, it will blinking for half a second and turn off in the same amount, repeatedly for 10 circles.

case 2: when the Arduino receives 2, it will blinking for 200ms and turn off in the same amount, repeatedly for 10 circles.

case 3: when the Arduino receives 3, it will blinking for a second and turn off in the same amount, repeatedly for 10 circles.

Youtube link:

<https://youtu.be/2j4tXAjkWi8>