Implementing Linux on the Zynq™-7000 SoC

Lab 3.1

Device Drivers and MIO GPIO

September 2012
Version 05

Table of Contents

Lab 3.1 Overview

This lab builds upon the skills covered in the previous labs.

Rudimentary user space interaction with the MIO hardware via sysfs is explored using the basic Linux system built during the 2.1, 2.2, and 2.3 labs.

Lab 3.1 Objectives

When you have completed Lab 3.1, you will know how to do the following:

- Use the sysfs subsystem to access MIO GPIO via the /sys folder

Experiment 1: Access MIO GPIO Hardware

With ZedBoard booted to the Linux command prompt, the MIO GPIO hardware can be accessed directly via the generic sysfs GPIO driver.

Experiment 1 General Instruction:

Boot ZedBoard using the SD card created in Lab 2.3 and interact with the MIO GPIO hardware. Note: If the ZedBoard is setup and still powered on from the previous Lab 2.3, skip ahead to Step 10.

Experiment 1 Step-by-Step Instructions:

1. Connect 12 V power supply to ZedBoard barrel jack (J20).

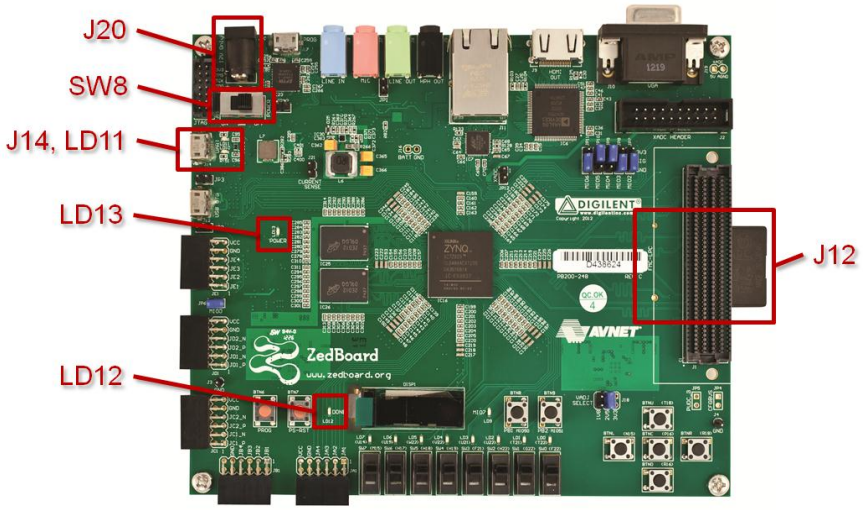

Figure 1 – ZedBoard Hardware Reference

2. Connect the USB-UART port of ZedBoard (J14) which is labeled UART to a PC using the MicroUSB cable.

3. Insert the 4GB SD card included with ZedBoard into the SD card slot (J12) located on the underside of ZedBoard PCB.

4. Verify the ZedBoard boot mode (JP7-JP11) and MIO0 (JP6) jumpers are set to SD card mode as described in the Hardware Users Guide:

   http://www.zedboard.org/sites/default/files/ZedBoard_HW_UG_v1_6.pdf

   A copy of the Hardware Users Guide is also located in the SpeedWay **C:\Speedway\Fall_12\Zynq_Linux\support_documents** folder for your convenience.

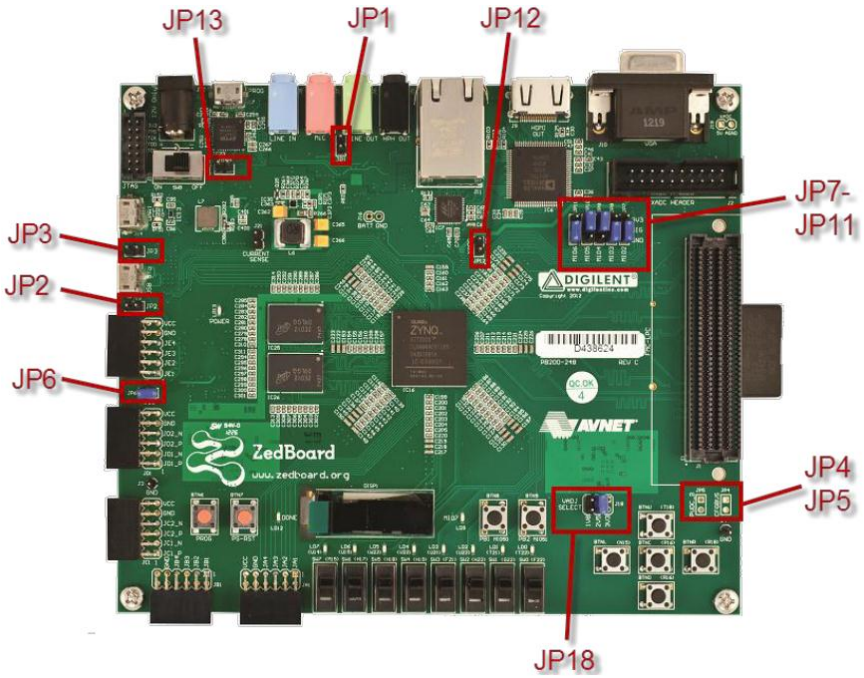

Figure 2 – ZedBoard Jumper Settings

5. Turn power switch (SW8) to the ON position. ZedBoard will power on and the Green Power Good LED (LD13) should illuminate.

6. Wait approximately 15 seconds. The blue Done LED (LD12) should illuminate.

7. On the PC, if a serial terminal session is not already open, open a serial terminal program. Tera Term was used to show the example output for this lab document.

Figure 3 – Tera Term Icon

8. If the amber USB-Link Status (LD11) does not flicker during boot to indicate activity, check the driver installation to determine if the device driver is recognized and enumerated successfully and that there are no errors reported by Windows.

9. Power cycle the ZedBoard and monitor the Tera Term window. When the terminal output from U-Boot and a countdown is observed, allow the countdown to expire.

Figure 4 – ZedBoard U-Boot Booting Linux

10. When the Linux command prompt is reached, take a look at the GPIO driver class within **/sys** subfolders.

    Notice how the GPIO driver exports controls via sysfs. Here we see that GPIOs are available for export via the export property.

```
$ ls /sys/class/gpio/
```

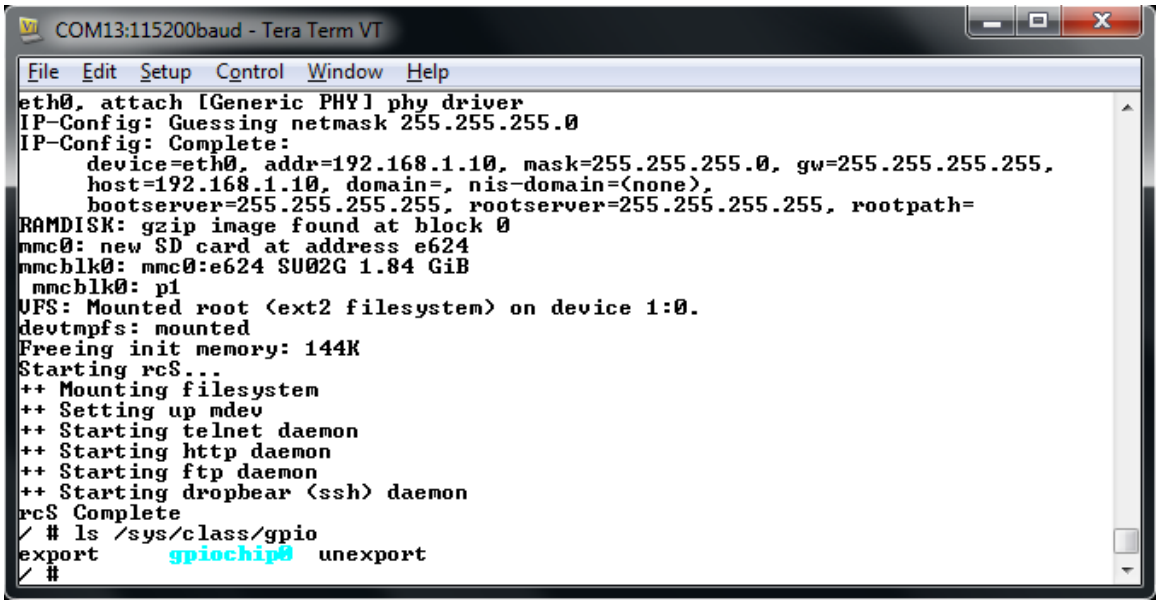

Figure 5 – Exploring the Sysfs Subsystem

11. Take a look at the ZedBoard schematic and determine which IO pin the MIO LED LD9 is connected to.

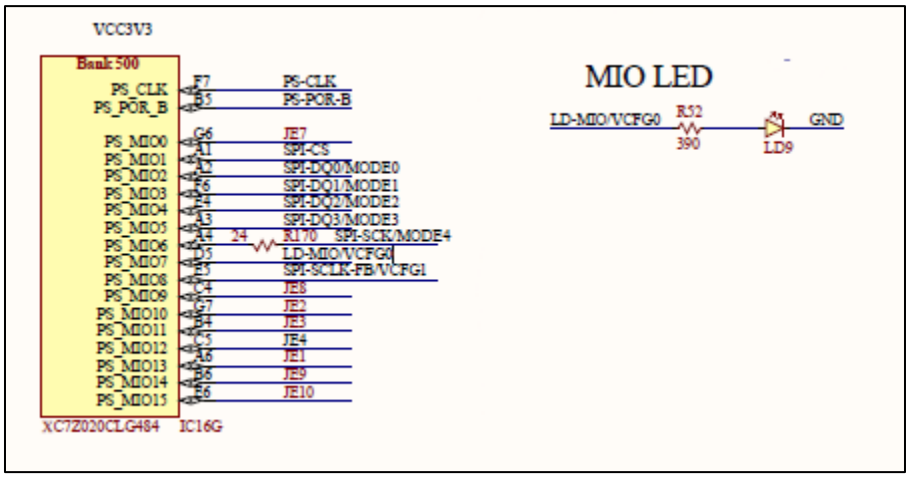

Figure 6 – ZedBoard Schematic Snippets Relating to MIO LED LD9.

12. In looking at the schematic, you should have determined that the MIO LED **L9** is connected to pin **D5** which corresponds to **PS_MIO7**. Using MIO number **7**, export the corresponding GPIO device to the sysfs file system so that the GPIO controls for **PS_MIO7** can be used.

    This is done by using the echo command to send the number **7** to the gpio device class **export** property.

    Then evaluate the GPIO folder again to verify that the new **gpio7** device has been exported to the sysfs file system.

```
$ echo 7 > /sys/class/gpio/export

$ ls /sys/class/gpio/
```

    Notice that the export property has caused the gpio7 node to become available. Behind the scenes, the GPIO driver received a write call and used the 7 parameter entry to determine which GPIO channel to enable and export control properties for. In the next steps, we will explore the function of the properties of the newly enabled **gpio7** node.

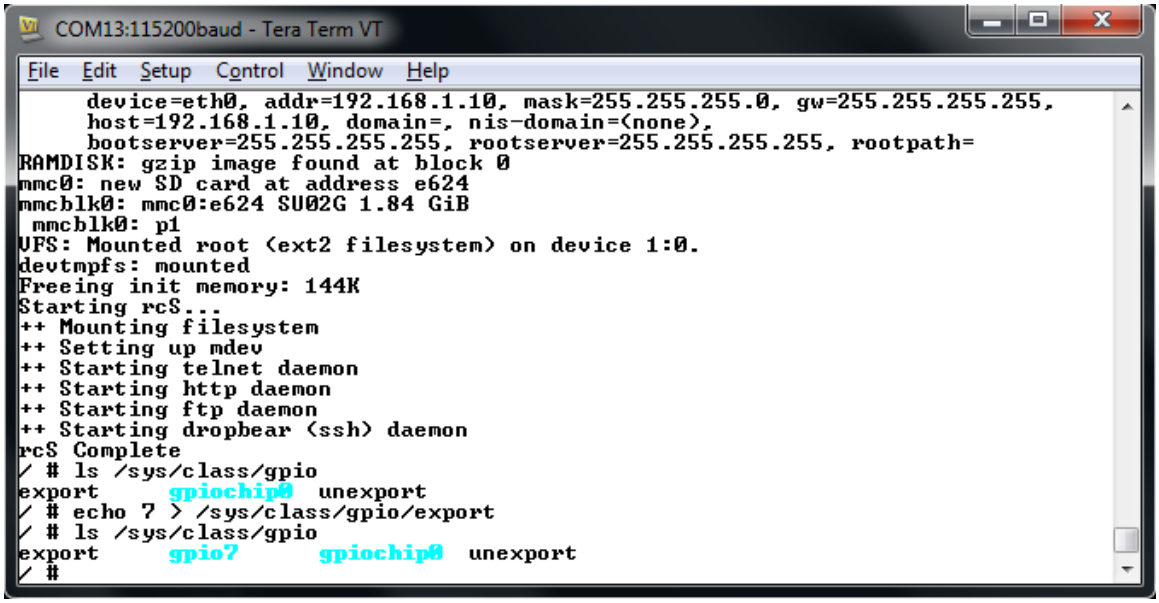

Figure 7 – Exporting GPIO7 Controls Via the Sysfs Subsystem

13. Evaluate the new **gpio7** node that was exported in the previous step.

```
$ ls /sys/class/gpio/gpio7
```

Notice that this node contains several properties which would normally be associated with a GPIO control.

Two of these properties are useful for this lab: the **direction** property and the **value** property.

The **direction** property is writable and controls whether the GPIO driver configures the controller for input or output. This property can be assigned either an **in** value or an **out** value.

The **value** property is read/writable and reflects either the output logic state of the GPIO when the **direction** property is set to **out** or reflects the input logic state of the GPIO when the **direction** property is set to **in**.

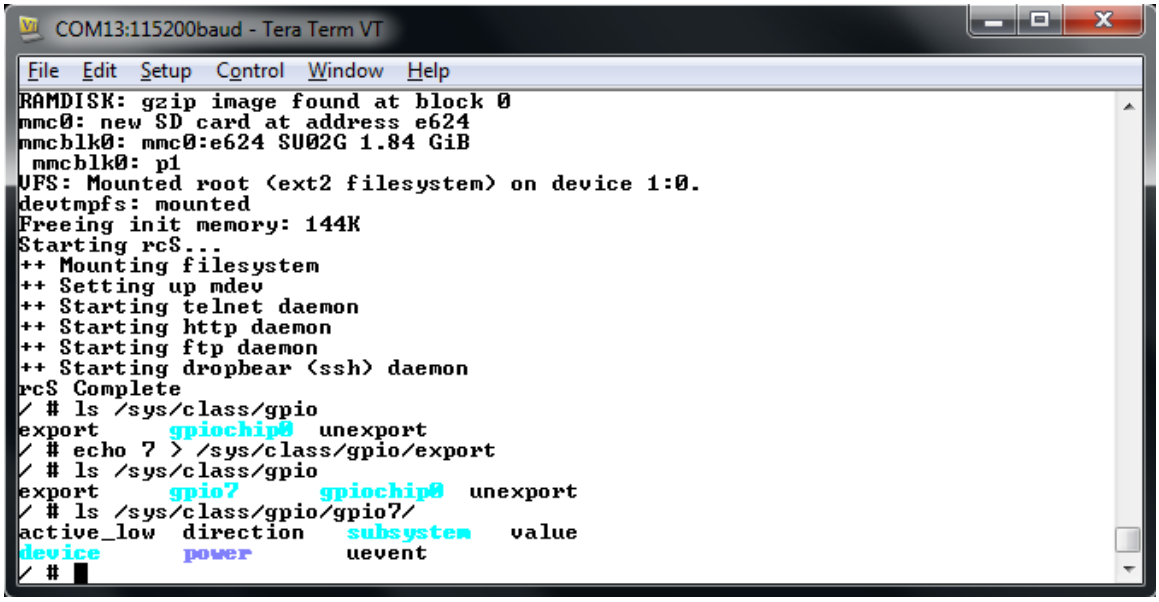

Figure 8 – GPIO7 Control Properties Via the Sysfs Subsystem

14. Modify the direction property of the gpio7 node and set it to an output.

```
$ echo out > /sys/class/gpio/gpio7/direction
```

15. Modify the value property of the gpio7 node and watch the ZedBoard LD9 LED as the command input is entered.

```
$ echo 1 > /sys/class/gpio/gpio7/value
```

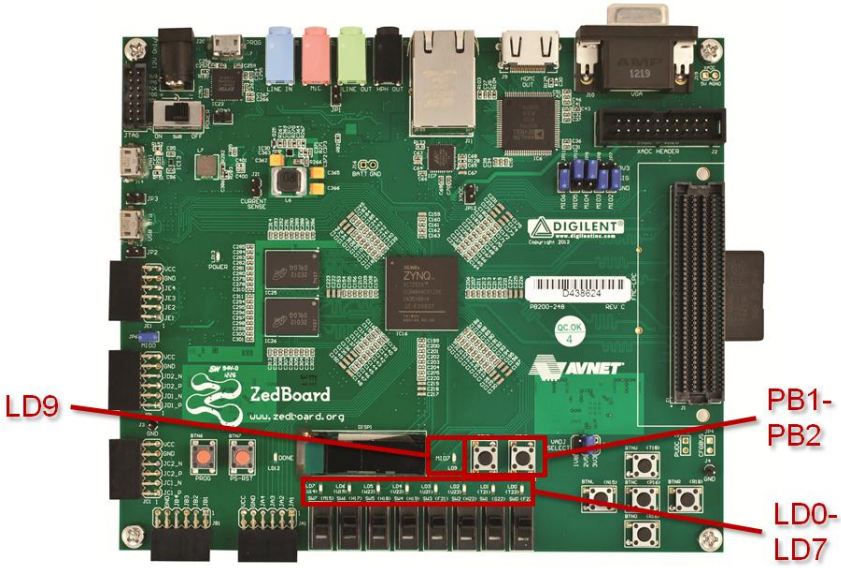

Figure 9 – ZedBoard Jumper Settings

Did you observe a change in state on LD9 LED?

Modify the value property of the gpio7 node again and watch the ZedBoard LD9 LED as the command input is entered.

```
$ echo 0 > /sys/class/gpio/gpio7/value
```

16. Continue experimenting with different inputs to the value. Which values are accepted, and which are ignored? How effective do you think it would be to implement a PWM control on this output using only software timing?

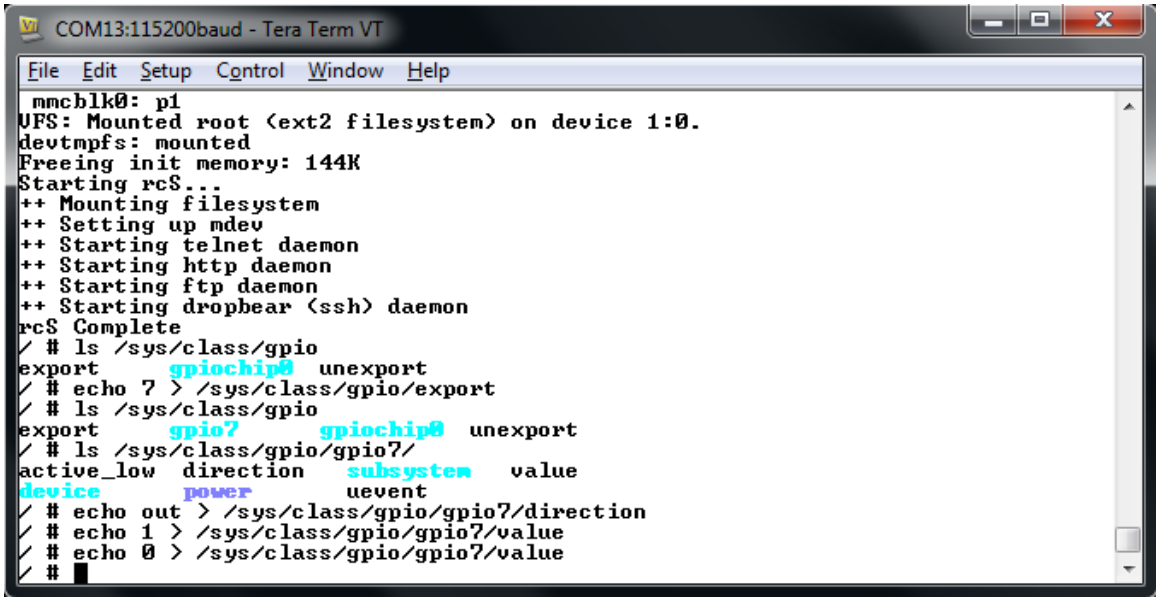

Figure 10 – Modifying the GPIO7 value Property

17. Perform a similar exercise using MIO push buttons **PB1** and **PB2** as input devices. Take a look at the ZedBoard schematic and determine which IO pins the MIO push buttons **PB1** and **PB2** are connected to.

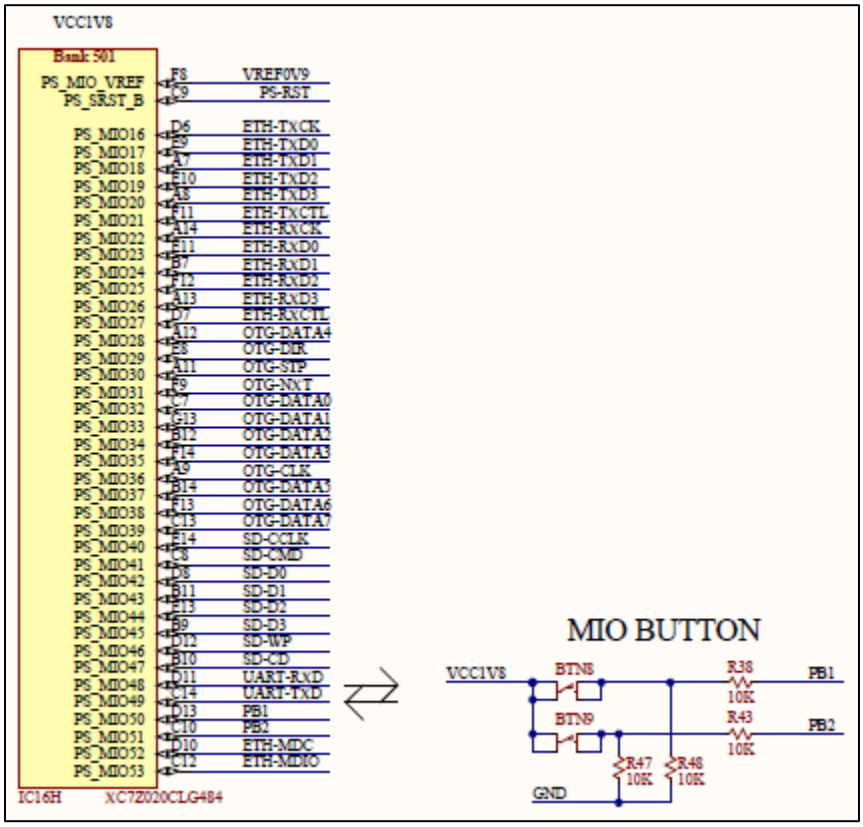

Figure 11 – ZedBoard Schematic Snippets Relating to MIO Push Buttons PB1 and PB2.

18. In looking at the schematic, you should have determined that the MIO push button **PB1** is connected to pin **D13** corresponding to **PS_MIO50** and that **PB2** is connected to pin **C10** corresponding to **PS_MIO51**. Using these MIO numbers, export the corresponding GPIO devices for use and evaluate the GPIO folder again.

```
$ echo 50 > /sys/class/gpio/export

$ echo 51 > /sys/class/gpio/export
```

19. Modify the direction property of the **gpio50** and **gpio51** nodes and set them to inputs.

```
$ echo in > /sys/class/gpio/gpio50/direction

$ echo in > /sys/class/gpio/gpio51/direction
```

20. Read the value property of the **gpio50** and **gpio51** nodes.

```
$ cat /sys/class/gpio/gpio50/value /sys/class/gpio/gpio51/value
```

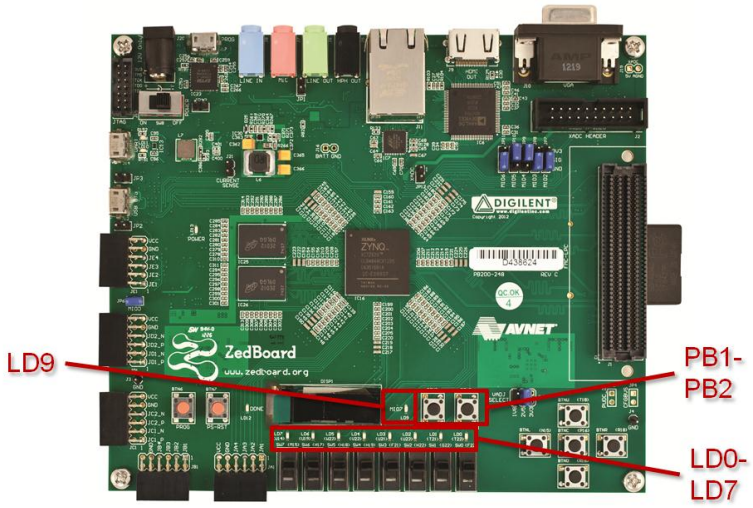

Figure 12 – ZedBoard Jumper Settings

Using the up arrow key on the keyboard to repeat a command in the command line history, repeat the above command while pressing one or both of the MIO push buttons. Did you observe a change in state on either of the value properties read from the push buttons?

21. Continue experimenting with reading the different input states from the value properties. Which values are accepted, and which are ignored? How effective do you think it would be to poll the push buttons for changes in state?

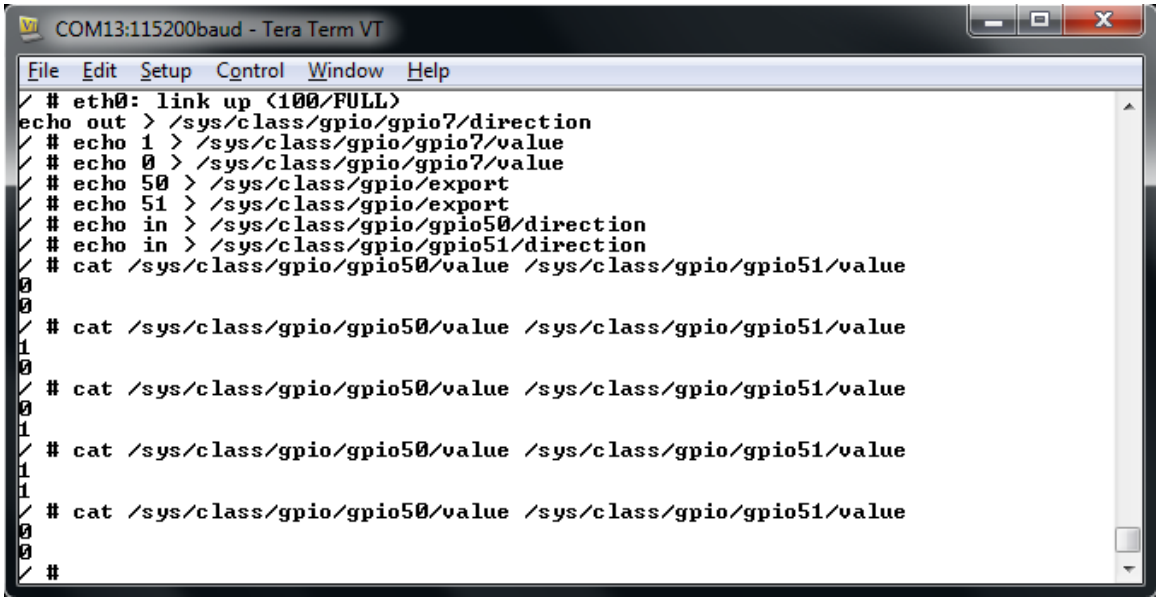

Figure 13 – Reading the GPIO50 and GPIO51 value Properties

Questions:

Answer the following questions:

- *Which GPIO number is connected to the PS MIO LED LD9?*

_____

- *Which GPIO numbers are connected to the PS MIO push buttons PB1 and PB2?*

_____

Exploring Further

If you have additional time and would like to investigate more…

- Write a short script to read the state of one of the two MIO pushbuttons PB1 or PB2 and write the state out to the MIO LED LD9.

This concludes Lab 3.1.

Revision History

Date	Version	Revision
17 Sep 12	00	Initial Draft
01 Oct 12	01	Initial Draft
19 Oct 12	02	Course Release
14 Jan 13	05	ZedBoard.org Training Course Release

Resources

http://www.zedboard.org

http://www.xilinx.com/zynq

http://www.xilinx.com/planahead

http://www.xilinx.com/sdk

Answers

Experiment 1

- *Which GPIO number is connected to the PS MIO LED LD9?*

MIO GPIO7

- *Which GPIO numbers are connected to the PS MIO push buttons PB1 and PB2?*

MIO GPIO50 and GPIO51