ECE353 In-Class Exercise

GPIO Pins – Analog Inputs

Problem Objectives

- Write routines to set the Analog Select and Alternate function registers of a GPIO port
- Write routines that will configure an Analog to Digital Converter on the Tiva Launchpad
- Read the Analog readings of the PS2 joystick and the potentiometer

1. Modify gpioPort.c

a. **Copy gpioPort.C from your digital GPIO exercise into this project (drivers/gpioPort.c).**

b. Add the following functions

```
// Configures pins an analog input pins
bool gpio_config_analog_enable(
     uint32_t baseAddr,
     uint8_t pins)

// Selects the alternate function for the provided
// pins
bool gpioc_config_alternate_function(
     uint32_t baseAddr,
     uint8_t pins)
```

2. Modify adc.c

a. Complete **`initializeADC.`** Look for ADD CODE in the comments.

3. Modify main.c

a. Complete the following #defines at the top of the file. You can determine the pins and ADC used by examining page 3 of the ECE353 Daughter board found on the class website.

```
// ADD CODE
// Define the base addresses for the GPIO port and
// the ADC attached to the PS2 joystick.
#define   PS2_GPIO_BASE    ?
#define   PS2_ADC_BASE     ?

// ADD CODE
// Define the Bitmask for each pin below
// Define the Bitmask for each pin below
#define   PS2_X_DIR_PIN      ?
#define   PS2_Y_DIR_PIN      ?

// ADD CODE
// Set the analog channel for each direction
#define   PS2_X_ADC_CHANNEL  ?
#define   PS2_Y_ADC_CHANNEL  ?
```

b. In the **`main`** function, configure the GPIO pins connected the X and Y direction of the PS2 controller to be analog inputs. Use the #defines above when configuring the pins.

c. Call the ADC initialization function. Use the #defines above

4. Open a Serial Debug Interface @ 115200.

a. Press each of the push buttons
b. The PS2 button is pressed by pressing the joystick down in the Z direction.
c. Verify the readings of the X and Y direction change as you move the PS2 joystick.

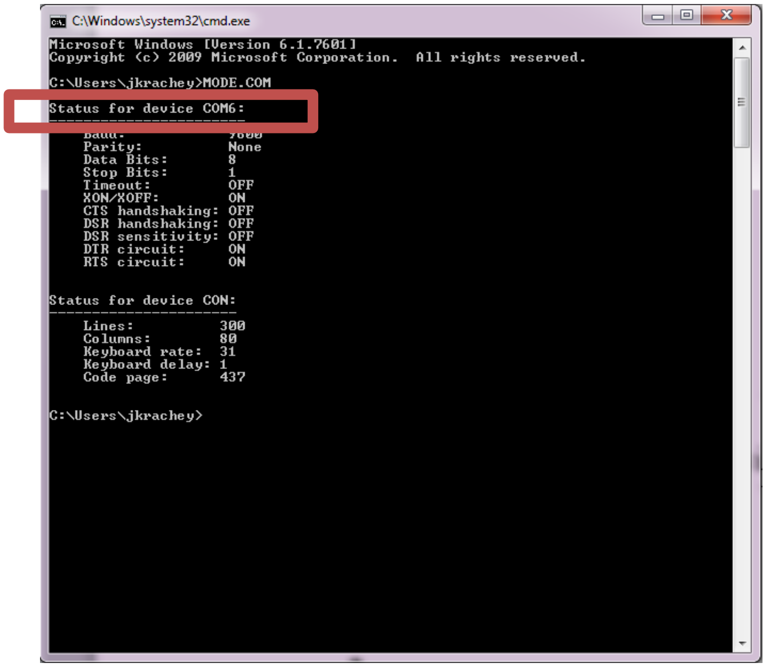
5. What to Turn In

Turn in **`adc.c`** to the dropbox on the course website.

Setting Up Serial Debug Port

We will use the serial debug port on the Tiva Launchpad so that you can use printf statements to display information to the user. In order to use the serial debug port, you will need to determine which COM port your Tiva Launchpad has been assigned by Windows.

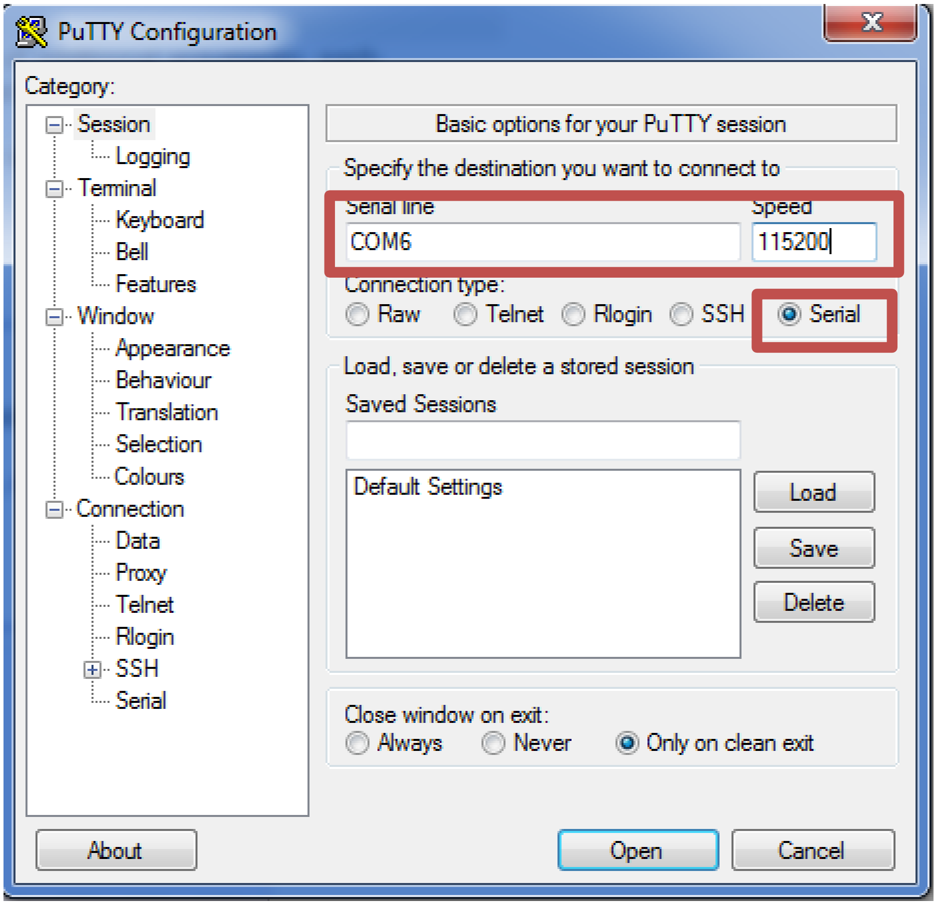
Open a windows command prompt issue the command **`MODE.COM.`** In the example below, we see that the Launchpad is connected to COM6. Your Launchpad will show up as a COM port greater than 3.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\jkrachey>MODE.COM

Status for device COM6:
-----------------------
    Baud:            9600
    Parity:          None
    Data Bits:       8
    Stop Bits:       1
    Timeout:         OFF
    XON/XOFF:        ON
    CTS handshaking: OFF
    DSR handshaking: OFF
    DSR sensitivity: OFF
    DTR circuit:     ON
    RTS circuit:     ON


Status for device CON:
----------------------
    Lines:          300
    Columns:        80
    Keyboard rate:  31
    Keyboard delay: 1
    Code page:      437


C:\Users\jkrachey>
```

Now that you have determined your COM port, open Putty. Click the Windows Start button, then type “putty” into the search box.

In Putty, click on the serial radio button, enter the COM port above, and set the baud rate to 115200.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: COM6

Speed: 115200

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit