

## **ECE 2534 – Spring 2017 – Lab 2**

### **The R2D2 Terminal – SPECIFICATION DOCUMENT**

**This lab is an individual assignment.**

#### **Introduction**

This laboratory exercise uses the LCD and the UART to display characters received from a terminal application running on your laptop. The application supports multiple display colors and baud rate settings, which can be selected by a button on the sensor board. By completing this assignment, you will get experience in the following.

- Configuration and use of the MPS432 UART Peripheral
- Configuration and use of a terminal application to test a UART connection
- Use of the LCD on the sensor board
- Reading and debouncing of sensor board buttons
- Use of the One-shot Timer to create fixed-length delay

This assignment is not a quick one-hour project. It requires you to study documentation and experiment with peripherals. IT IS MANDATORY TO START THE ASSIGNMENT AS SOON AS YOU RECEIVE IT.

#### **Honor Code Requirements**

You must complete this lab *individually*. Do not discuss your solution with anyone except for your instructor or a GTA. (You are encouraged to seek help, if needed, from any GTA in the Computer Engineering Lab.) Consider all information that you generate to be proprietary. Copying or using any element of any other person's implementation or solution is a violation of the Virginia Tech Honor Code.

#### **Hardware Components**

To complete the assignment you will need the following hardware components.

- MSP432P4 Launchpad with BOOST-XL Sensor board
- Laptop with USB connection to MSP432P4 Launchpad

#### **Software Tools**

To complete the assignment you will need the following software tools

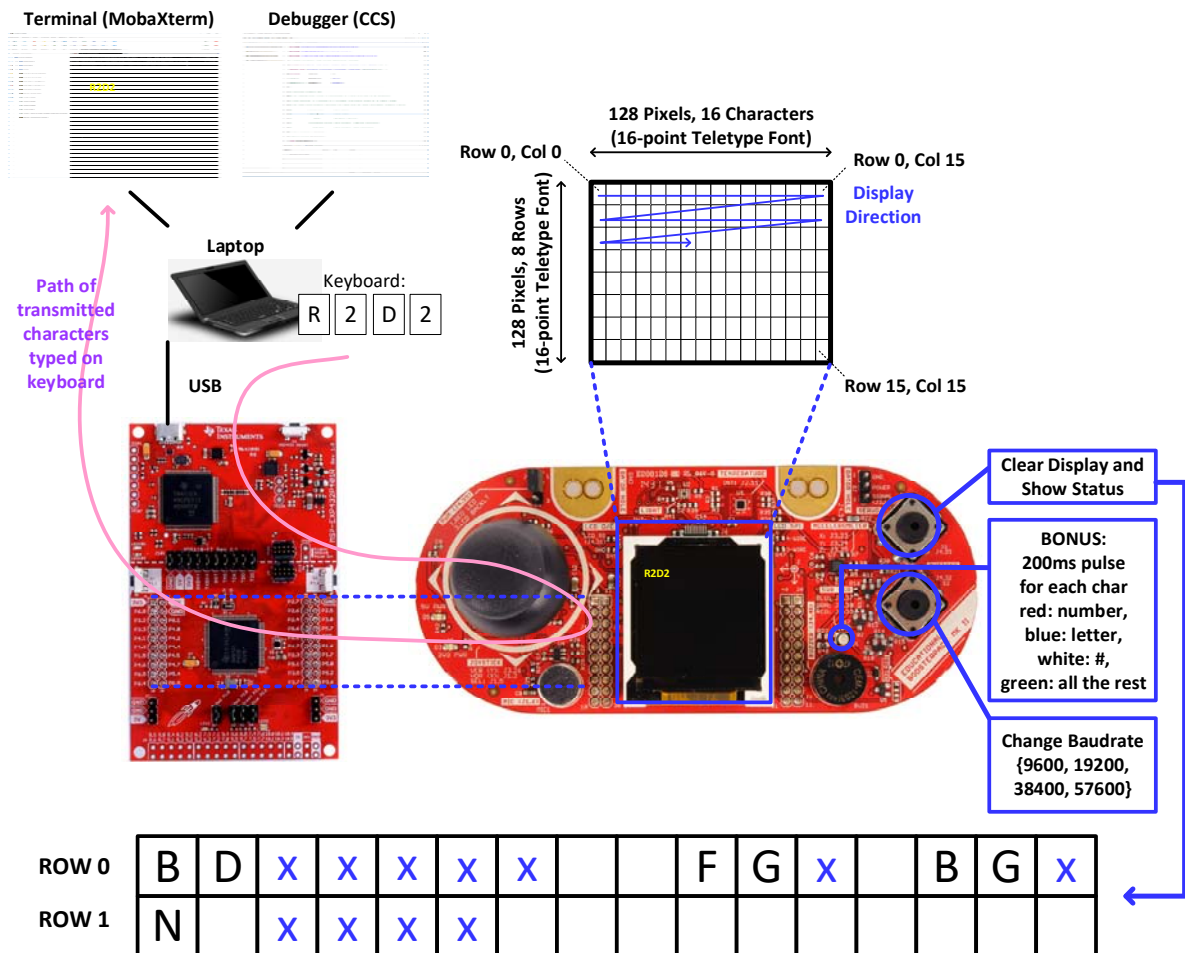
- CCSv7 (as installed in Lab 1)
- MobaXterm (a terminal application, see <http://mobaxterm.mobatek.net/>)

#### **Documentation Manuals**

To complete the assignment you will need the following documentation

- TI Driverlib Software Library Manual
- TI Graphics Library Manual
- Sensor board and launchpad User Guide (for connectivity of the Switches and color LED)
- MSP432P4 User Guide (for a description of the UART operation)

## Lab Specifications



You have to develop a terminal application for the MSP432P4. Here is a description of the required functionality.

- A terminal application on your laptop, called MobaXterm, connects to a UART on the MSP432P4. When you type a character on your laptop keyboard, it is transmitted through MobaXterm and over USB to the UART of the MSP432P4. The MSP432P4 software reads the character from the UART receiver register and echoes it back to the laptop by retransmitting it over the UART connection to the MobaXterm application. The returned character will then show on the MobaXterm application on your laptop.
- Each received character is also displayed on the LCD screen, a 128 pixel by 128-pixel color display on the sensor board. The LCD is connected through the MSP432P4 through a peripheral (SPI) which we will study at a later time. In this lab, we will make use of a software library that provides high-level access to the LCD screen. For character display, the LCD screen is partitioned into eight rows of 16 characters each, and a fixed-size point-16 font is used to display the characters.
- The LCD screen is written line by line from the top-left character position to the bottom-right character position. The LCD does not scroll when the bottom-right character position is reached. Rather, the display ‘wraps around’ and continues on the top-left position,

- overwriting any characters that were previously there.
- The two buttons on the sensor board each serve a purpose.
  - The top button erases the display and writes a status message on the first two lines of the display. The contents of the status message are discussed under ‘Status Message’ further in the document. After the status message is displayed, printing of received characters continues on the leftmost position of the third line of the LCD.
  - The second button sets the baud rate of the UART to one of four possible speeds: 9600, 19200, 38400 or 57600 baud. The default baud rate is 9600 baud. After pressing the lower button, the baud rate changes to 19200 baud. After pressing the lower button again, the baud rate changes to 38400 baud. After pressing the button again, the baud rate changes to 57600 baud. After pressing the button a fourth time, the baud rate cycles back to 9600 baud gain. Your software should properly debounce this button.
- The character stream received on the MSP432P4 is analyzed for ‘inline commands,’ which are indicated by a hash character ‘#,’ a single letter, and an optional argument. Inline commands are not printed, but they change settings on the application. Your program has to support the following list of commands.
  - **#fn** sets the foreground color for the LCD to *n*, with *n* a number from 0 to 7 representing a color
  - **#bn** sets the background color for the LCD to *n*, with *n* a number from 0 to 7 representing a color
  - The color codes are as follows: 0=black, 1=red, 2=green, 3=yellow, 4=blue, 5=magenta, 6=cyan, 7=white.
- **BONUS:** You will receive bonus credit if your implementation supports the following feature. The color LED turns on for 222 ms after each received character. The color of the LED depends on the type of character received.
  - The hash character # displays a white LED
  - A number (0 to 9) displays a red LED
  - A letter (a to z, A to Z) displays a blue LED
  - Anything that does not fall into the above categories displays a green LED.

## Status Message

The status message displays the basic settings of the application. The exact format of the status message is shown at the bottom of the figure on page 2. Here are a few examples.

```
bd 9600 fg5 bg2
n 0028
```

This message tells that the baud rate is 9600 baud, the foreground color on the display is magenta (fg5), and the background color is green (bg2). Also, at the time the message was displayed, 28 characters have been received from the laptop.

```
bd19200 fg0 bg7
n 0121
```

This message tells that the baud rate is 19200 baud (bd19200), the foreground color on the display is black (fg0), and the background color is white (bg7). Also, at the time the message was displayed, 121 characters have been received from the laptop.

When the status message is generated, the LCD Display will also be cleared, and the LCD continues printing on line 3 of the LCD.

## Running the starter code

- Download the Lab 2 starter code and run it on the board.
- Start MobaXterm and connect to 'COM3 XDS110 Class Application/User UART'. It's possible that COM3 will be another number (e.g. COM5). In that case, choose the one that has the 'XDS110 Application/User UART'. Select 9600 Baud. In 'advanced settings,' verify that you have 8 data bits, one stop bit, no parity bits, and no flow control.
- When the application starts, the LCD screen turns blue, and at the bottom of the display there's a message ('R2! WHERE ARE U?').
- On MobaXTerm, you see the message: 'Hello!'. When you type on the keyboard from the MobaXterm application, you will see the characters being echoed back. Each time the MSP432 received a character, the red LED on the Launchpad board will toggle.

## What to turn in

- Before the deadline, push your repository back to GitHub.
- Write a short report with the following
  - **Section 1:** Description. A short section (a paragraph or two) describing what your application can accomplish. Ideally, that matches the Lab description. However, use your own words to describe it. If your application cannot perform all the tasks or performs some of them suboptimally, you need to mention them in this section.
  - **Section 2:** Finite State Machine Design. Explain how you designed the Finite State Machine(s) in your program. Explain what they are used for, and explain what event sequence they are decoding.
  - **Section 3:** HAL Design. Give a list of the low-level functions you have developed to simplify the development of the main program and explain the purpose of these functions. You don't have to list the implementation of the functions; you only have to provide the header, and describe what the functions do.