ECGR 4101/5101, Fall 2016: Lab 3

Testing the Renesas RX63N Demonstration Kit (YRDKRX63N)

Version 1.1 – 9/22/2016

Learning Objectives:

This lab will allow you to validate that your RX63N demo board is functional by demonstrating how to create a new project in HEW and load the compiled executable onto your RX63N Demo board.

General Information:

- Attempt to power up the RX63N demo board and observe the pre-loaded test program. If the board is not pre-loaded (*id est*, no LED's are blinking), you will have to download a program to the RX63N board *via* HEW by following the RX63N quick start guide.
- Follow the steps provided by the lab 3 supplemental information document and load the RX63N tutorial project onto the RX63N board.
- Modify the tutorial project to perform the specified requirements
- Build the project and load it onto your RX63N Board. Run the program and observe the operation.
- Demonstrate your working project to the TA, and turn in a lab report.

Getting Started:

You may use the PCs in EPIC 2130 or your own PC to do this lab experiment. If you want to work on lab assignments on your own PC, then you will need to load the necessary tools on your PC in order to perform this exercise.

Test your board by attaching it to a computer *via* USB. Ensure the LCD lights up and if the original Renesas test program is loaded that lights are flashing and text is displayed on the screen.

Laboratory Assignment Overview: A Model Train in the Digital Age…

"After four years of studying to be an engineer, I wonder when they are going to let us drive the train?"
- *Unknown senior overheard in the hallway.*

Model trains are a great source of amusement for both kids and adults alike and what better way to dive into embedded systems than creating your very own digital model train!

For those of you who have never played with a model train set before or have just forgotten, a typical model train set comes with small conductive rails, a small electric locomotive that rides on top of the conductive rails, freight cars or passenger cars that are pulled by the locomotive, and a small power supply that is connected to the conductive rails to power the locomotive. A typical train power supply has the ability to change the rail polarity in order to change the direction of the train and a variable voltage source to vary the speed of the train.

In order to build a digital model train we will substitute the conductive rails and train with the circular LEDs ring on the RX63 demo board and use the RX63 demo boards buttons to control the train's direction while using the RX63 demo boards potentiometer to control the trains speed. We can also use the RX63 demo boards LCD display to write information about our digital train for easy viewing.

Requirements:

- The code generated is written in C for the RX63N Evaluation Board.
- The code is well commented and easy to follow.
- LED 4 to LED 15 symbolically represents the model train track. Note there are 12 LEDs total or 12 track segments in our model railroad.
- When a LED is off it means that there is no train on that section of track.
- When a LED is on it means that either a locomotive or train car is on that section of track.
- Our model railroad will have a total of 5 cars on the track and this implies that at any given time there should be always be only 5 LEDs actively emitting light.
- When SW1 is pressed the train will move forward and the LEDs will move in a clockwise motion.
- When SW2 is pressed the train should stop and the display should show the stopped station (the center car of the train).
- When the train is stopped, the green and red LEDs should alternately blink.
- When SW3 is pressed the train will move backwards and the LEDs will move in a counterclockwise motion.
- The LCD should say what direction the train is currently "FORWARD" (when moving clockwise), "REVERSE" (when moving anti-clockwise) or report that the train has "stopped".

Laboratory Assignment:

- Review the instructions given by the RX QuickStart guide. Note carefully the steps required to build a program, download, and debug a program.
- In the “Welcome” window of HEW, choose “Browse to another project worskspace”. Go to C:Micrium->Software->EvalBoards->Renesas->YRDKRX63N->HEW->uCOSIII. Select uCOSIII.hws. Immediately select “file->save the workspace as” and add a prefix of “train_”. Build the newly created project and download it on the board. Note the change of behavior of the board. (if you get complier errors at this point you will need to review the Lab 1 Supplemental Information document for help on fixing them)
- In the project editor, choose app.c as the file to edit. Include the “platform.h” header file.
- You will need to change the program in a few places to satisfy the project requirements:
    - At boot-up, instead of showing "Renesas " on the LCD, show your first name
    - At boot-up, instead of showing "RX63N" on the LCD, show “Train Demo”
    - At boot-up, Add an additional LCD message to the fourth LCD line that says “Mode:STOP”
    - Fifth line of LCD should display “STATION:<number>”.

  Download and run this new code. Note the change of behavior of the board.

    - To Download and run code, you must first connect your RX63N board to HEW. First, be sure to change the Debug parameter from "DefaultSession" to "Jlink." You will see this at the top right of the first toolbar in HEW.
    - Then go to Debug -> Connect to connect the board. HEW will then prompt you for various parameters, but the default ones are fine, so simply click next through the windows that pop up.
    - Once you have connected the board, be sure to build your code by going to Build -> Build All.
    - After the code is successfully compiled, go to Debug -> Download Modules, and select the first option, which should be your compiled code.
    - Once the code has been downloaded, go to Debug -> Go to run the code.
- You will notice the function BlinkyTask as the function controlling the LEDs. You will need to change the functionin a few places to satisfy the project requirements:

- At boot-up BlinkTask() is executed, Modify this function such that it will only exit this function when either (SW1,SW2,SW3) is pressed (hint: use register named PORT4.PIDR.BIT.B0/B1/B4 instead of the keyword SW1, SW2, SW3 respectively)

Download and run this new code. Note the change of behavior of the board.

- Update the LED_On() function in bsp.c.
  - Update the function to turn on/off LED4 to LED15 (hint see the existing LED_On() and LED_Off() functions in bsp.c. Be aware that the existing function can only modify 1 port at a time.
- Continue updating blinkyTest() function
  - Initialize a variable named "start" with a value ranging between 4 and 15. LED_On(start) will light up 5 consecutive LEDs that correspond to the location of the train. For example, if the number 4 was passed into this function then LED4, LED5 LED6, LED7, and LED8 should light up. LED4 would correspond to the train locomotive while LED8 would correspond to the train caboose. To provide another example, if the number 15 was passed into this function then LED15, LED4, LED5, LED6, LED7 should light up while if the number 14 was passed into this function then LED14, LED15, LED4, LED5, LED6 should light up. This function should turn off all LEDs that are not used to represent the train's location. (hint be sure to use the functions you created earlier to help you in this task).
  - Your code should start with the train stopped and polling for the user to press SW1, SW2 or SW3. Table 1 summarizes the requirements when any of the 3 switches are pressed.

Switch	**LED Function**	**Third Line of the LCD Display (note that there are no spaces)**
1	Train moves Clockwise	MODE:FORWARD
2	Train Stops	MODE:STOP
3	Train moves counter-clockwise (for you who speak "British-English", anti-clockwise)	MODE:REVERSE

TABLE 1

  - Download and run this new code. Note the change of behavior of the board. (at boot-up you should now have 5 LEDs on at the current train location and all the LEDs should blink after you press a button).

To Demo and Submit:

To submit, have the demonstration sheet below printed off. Demonstrate your working (or partially working) code to the TA. Upload a text file to Canvas containing all the modified functions in alphabetical order.

Embedded Systems Lab Demonstration Validation Sheet

This sheet should be modified by the student to reflect the current lab assignment being demonstrated

Lab Number:	Lab 3
Team Members	Team Member 1: Team Member 2:
Date:	

Lab Requirements

1	LCD displaying your name and “Train Demo” on correct lines	
2	LED are displaying a 5 car train that is stopped at boot-up	
3	LCD displaying the trains current direction is stopped at boot-up	
4	Pressing SW1 makes the train move forward and the LCD displays that the train is moving forward (LED move correctly).	
5	Pressing SW2 makes the train stop and the LCD display that the train is stopped and stopped station while the green and red LEDs alternate	
6	Pressing SW3 makes the train move backwards and the LCD displays that the train is moving backwards (LED move correctly).	