**Assignment #1**

**- Control of a Train using Tornado development environment and Wind River RTOS -**

Members of Group 3:

* Mark Baker
* Gary Steelman
* Melyssa Fowler
* Jie Fang

1. **The procedure we used to develop the program:**

* **Set up Tornado development environment and Wind River RTOS:**

**Step 1:** Open the FTP server and then the RTOS “vxworks1 new” in Tornado 2.2. Check if IP address and subnet of the RTOS are set up correctly, and the IP of the board as well.

**Step 2:** Turn on the board and make sure “ready” displays on the RTOS.

**Step 3:** Open the Tornado development environment. Configure and launch a new server using the target server name “Project1” and IP address 192.168.94.2. Close the old server and start with this new one (e.g. “Project1@cs58220” in the available server list at the top left).

**Step 4:** Create a downloadable application module for the RTOS as follows:

(1) Set a toolchain “PPC860gnu”.

(2) Add a project, named “Project1\_GP3”, at a specific location to a new or existing workspace (e.g., “c:\PGMJ\PPC860gnu\Workspace0.wsp”).

(3) Add our program file (e.g., “Project1\_GP3.c”) under this project.

(4) Compile the program file and an object module “Project1\_GP3.o” is created/updated.

(5) Build the program file and an object module “Project1\_GP3.out” is created/updated.

**Step 5:** Download the module “Project1\_GP3.out” to the board. Type i command to launch the shell on the RTOS. Then use commands to control the train by calling the init() function first.

* **Develop/debug our program:**

We edited our program file in Step 4(3) and made changes if any. For each command that we wished to implement, we looked up the binary code in the instruction manual. We translated this number into hexadecimal and wrote it to a buffer that was sent to the train. While debugging, we needed to repeat Step 4(4) & 4(5) and Step 5 to check the result. As real-time system development, we also had to care about the train’s response time and behavior with respect to each command.

1. **Differences between this procedure and the development of a general purpose application:**

* The program has a different running environment, instead of running on a host machine, it has to be downloaded to a board after being compiled.
* There is no main function required as the entry point into the program. Instead the init() function is called from the shell.
* The program does not keep track of the train’s current state.
* When the train was not responding to the commands from the program, we had to shut down the train system mechanically. For normal program, you can just kill the process.
* When the program exited without explicitly sending a stop command to the train, the train kept going.
* The c version the compiler supported is slightly pickier than other compilers. It doesn’t like the // comment.