I considered this FanController will be embedded in robot and running in robot machine to monitor the temperature and control the duty-cycle of the fan. So I assume that once the robot is turned on and running, so do the Fan and the FanController software.

Test plan: To test the functionalities of the Fan Controller. testFanController.java is a test program to run the FanController.java

Test case1: Test that duty cycle should be 20 if the temperature is less than or equal to 25 degree C.

Test case2: Test the duty cycle should be 100 if the temperature is larger than or equal to 100.

Test case3: The duty cycle should be ((current-temperature – 25)(80/50) + 20) if the temperature is larger than 25 degree C or less than 75 degree C.

I created 1 java project and 1 java maven project. Both are the same code with 2 different project structures. I exported these 2 projects to jar files and put them in FanControllerJars.zip file. You can install eclipse and import these two packages jar files and run testFanController.java. I’ll also attach these 4 java files with the email, in case of you did not use eclipse to import it. Then you save them in a folder and compile all 4 java file with javac and run it testFanController.class in command-line. I’ll try to convert it and re-write in it either python or C++ and send to you later in this week. I’ll try to work on it on Memorial day, since I am still off for Monday.

1. FanController.jar
2. FanControllerMavenProject.jar

I expected 2 APIs on the robot to be implemented:

1. FanController.java: I implemented this FanController.java. It is the fan controller software engine to get the current temperature from thermometer and calculate the duty-cycle. Then FanController will send the duty-cycle integer number to Fan to control the fan duty-cycle. It is supposed to be running when the robot was turned on and stop when the robot was turned off.
2. CurrentTemperature.java: FanController will call this API to get the current temperature reading. CurrentTemperature.java will be implemented to get current C degree from thermometer device. Currently, I mock the test data for testing purposes and It generated random float degree from 0 – 100. So it’ll test all the test cases that I layed out earlier in this document.
3. DutyCycle.java: I mock this API for testing purposes. This DutyCycle.java API will send the integer dutycycle (20-100) to the Fan device.
4. testFanController.java: It is a test program to test and run FanController.java.

It’ll be a good challenge for me if I have to write the API DutyCycle.java to send dutyCycle signal to Fan and control the fan. Also if I have to write CurrentTemperature.java to get reading from thermometer/thermostat to get the current temperature of robot. It’ll be very exciting for me to explore into this new hardware control mechanism if I have a chance to implement these two API.

In conclusion, I’d like to thank you for giving me a chance to work on this project. I am really enjoy working on this project for my Memorial long weekend. If you have any questions regarding how to run it and also about the logic calculation of the dutyCycle, please send me an email or give me a call, and let me know.