**Objectives:**

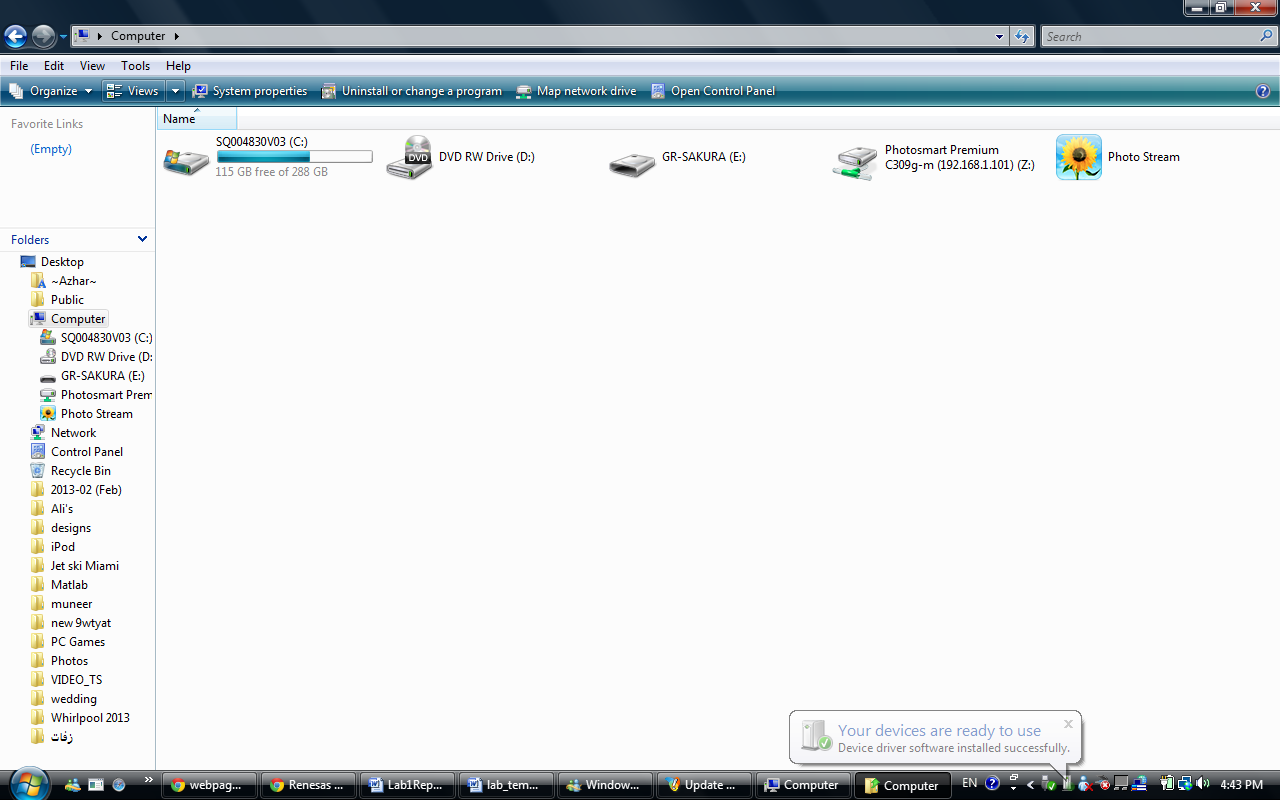
The objective of this lab was to understand and generate a project using the GR-Sakura web compiler; a code was compiled and built using the Renesas Sakura.

**General Steps:**

1. Go to the [web-compiler](http://update.renesas.com/gur/compiler.do?&languageCode=en)
2. Power up the Renesas Sakura board by connecting it to the computer via mini-USB
3. A new project was created, the code of Lab 1 was copy and pasted form Moodle into the project
4. The code was compiled and uploaded onto the Renesas Sakura board

**Detailed Steps:**

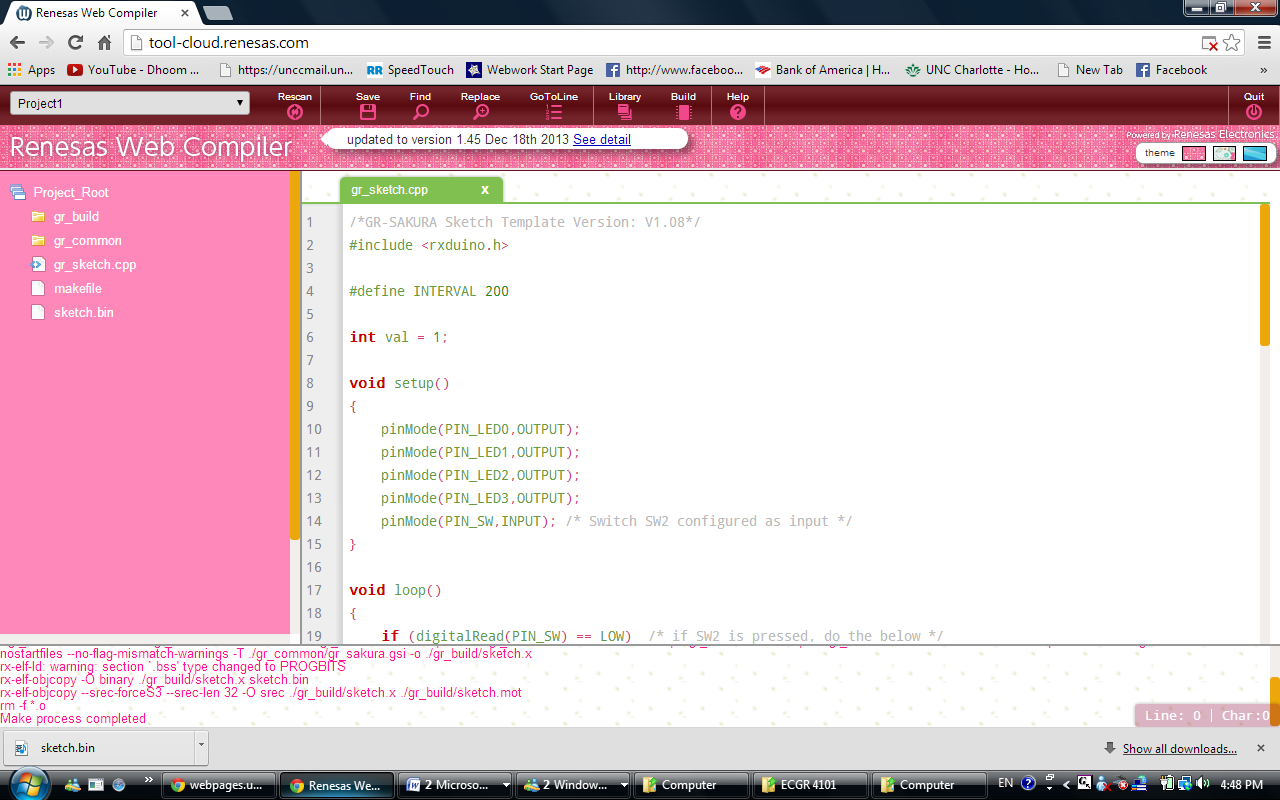
1. Press guest login and accept the condition to built the project 
2. Press the reset button and a there will be a file of the board name in my Computer file



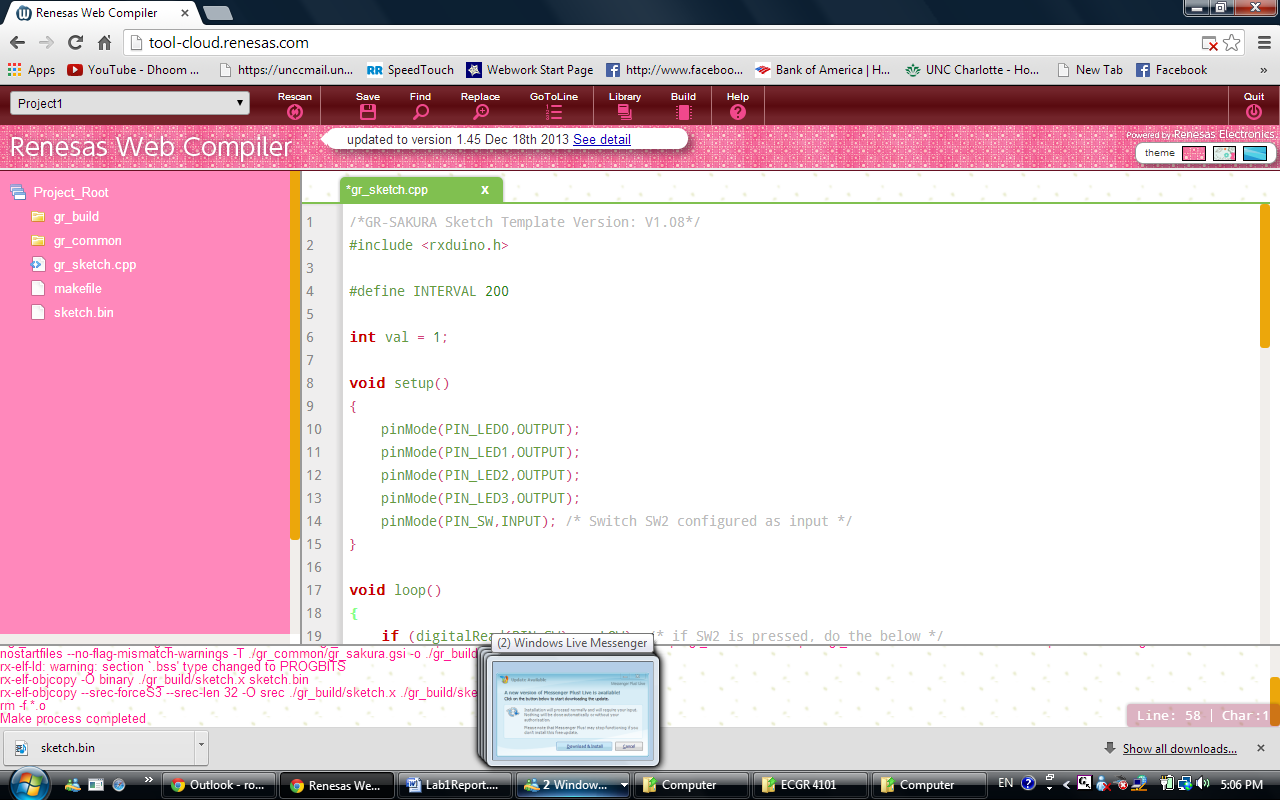
1. Copy the code of Lab 1, located on Moodle, and paste it into the gr\_sketch.ccp
2. Press the build button
3. After the build is complete a file called “sketch.bin” will be added to the project
4. Download “sketch.bin” file onto the GR-SAKURA(E:)
5. The code will run without pressing SW2
6. Press SW2 and see the difference

**Code Modified:**

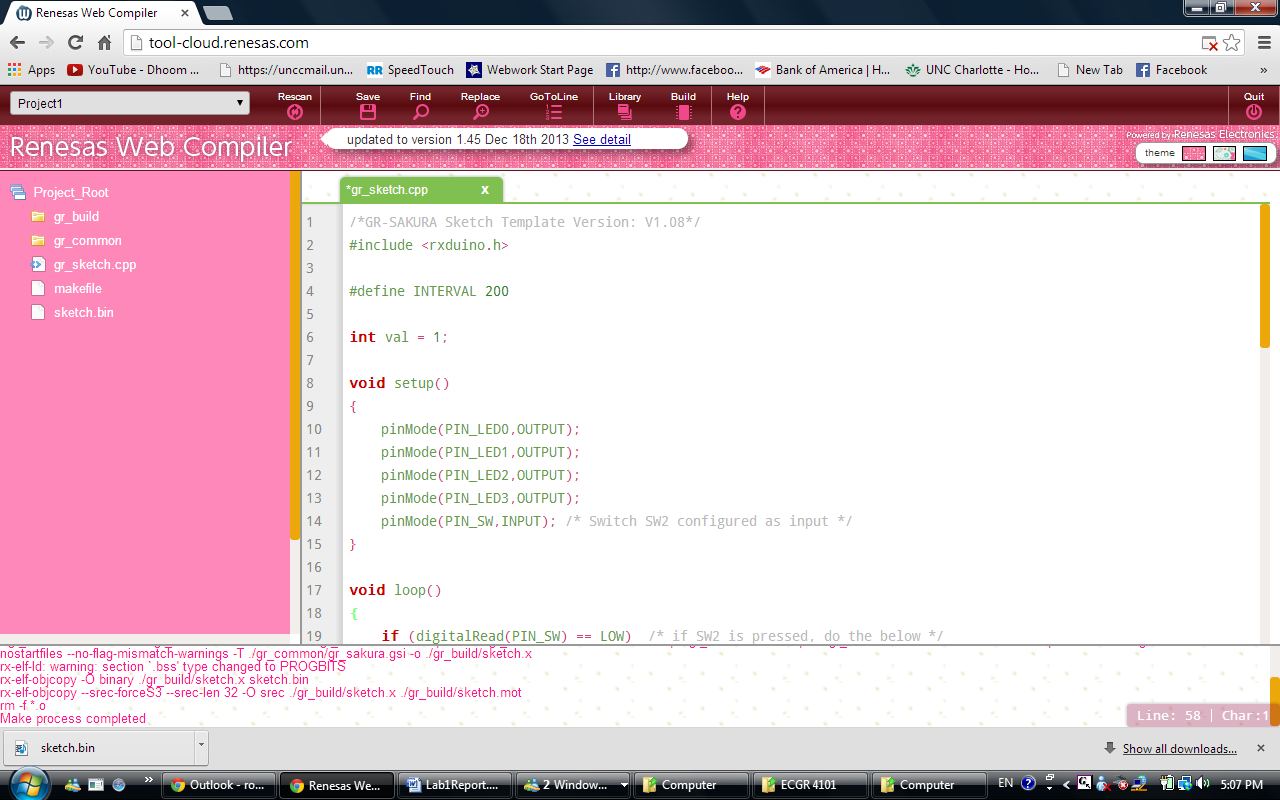
1. Calls the library header file and defines an interval which will be used to place a delay after the pin is activated



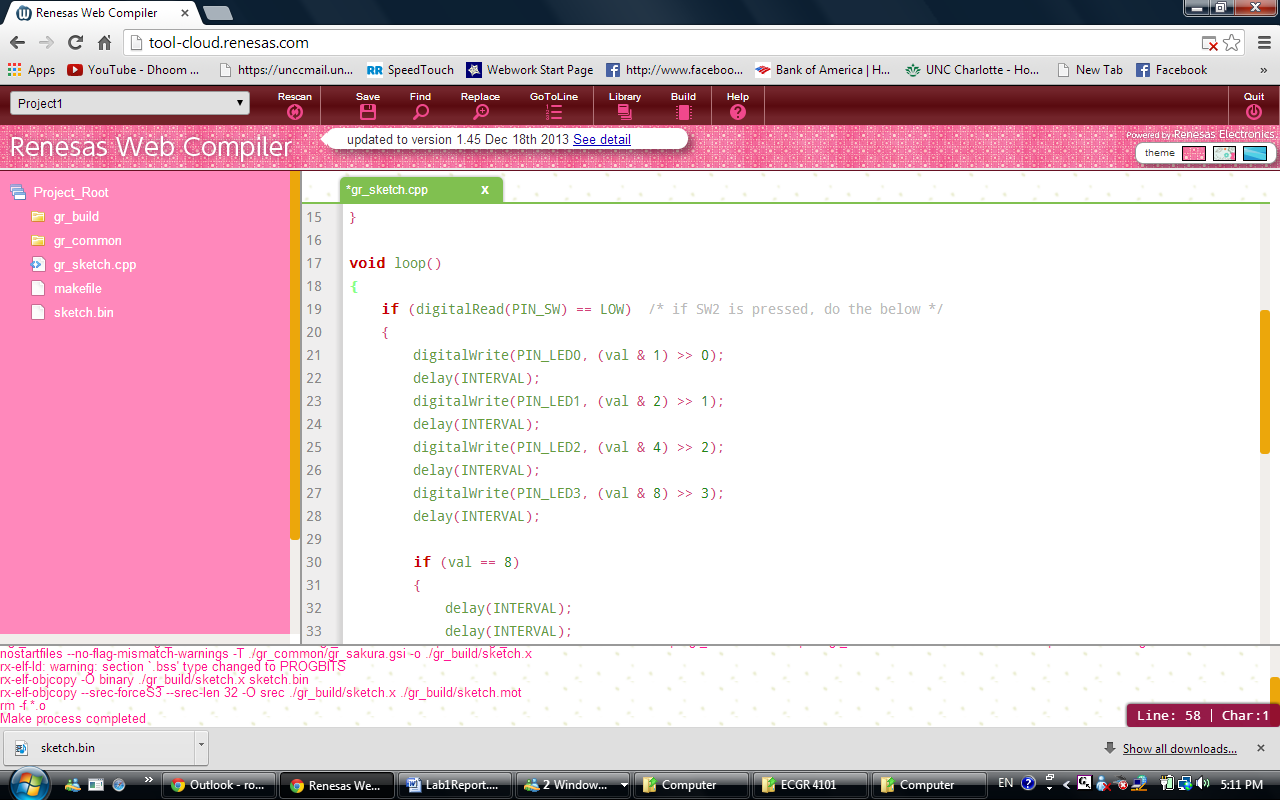
1. variable used for determining state of LED pins



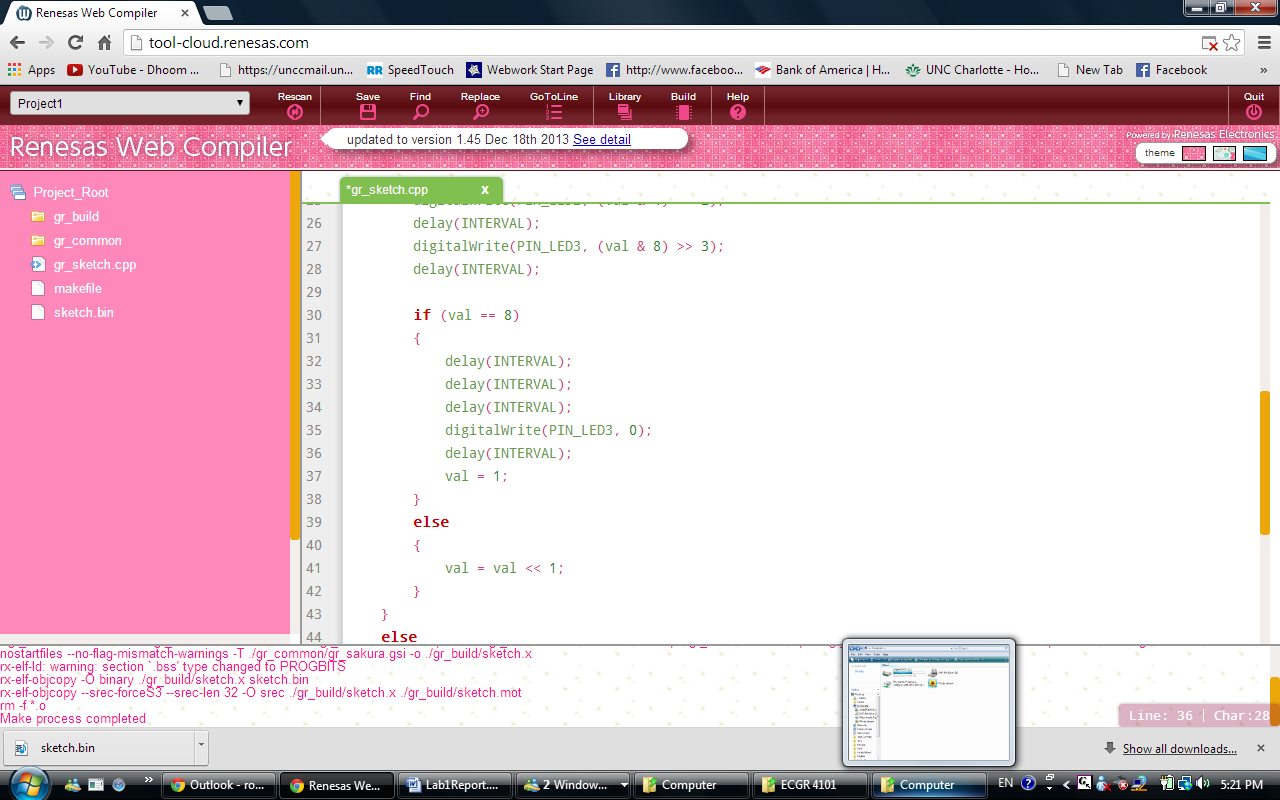
1. Sets the pins from LED0 to LED3 to outputs, and sets the switch (SW) as an input

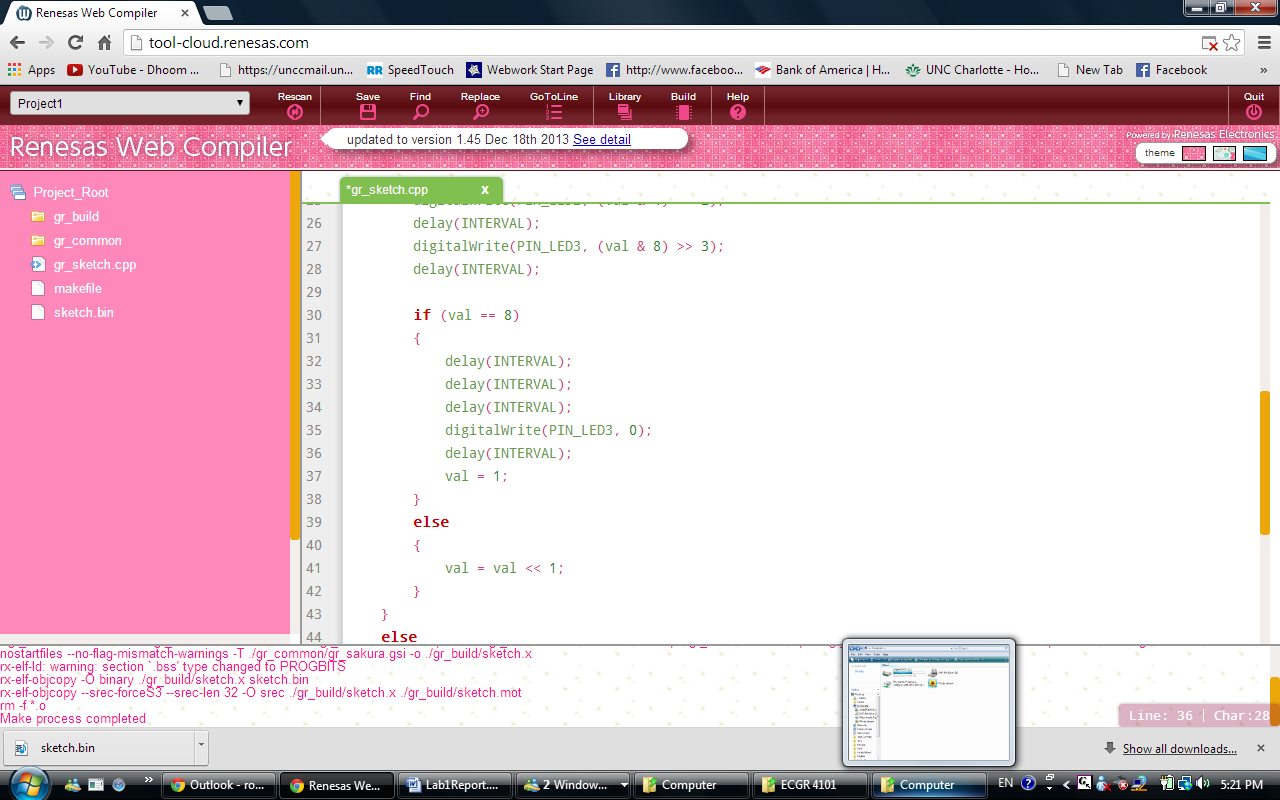


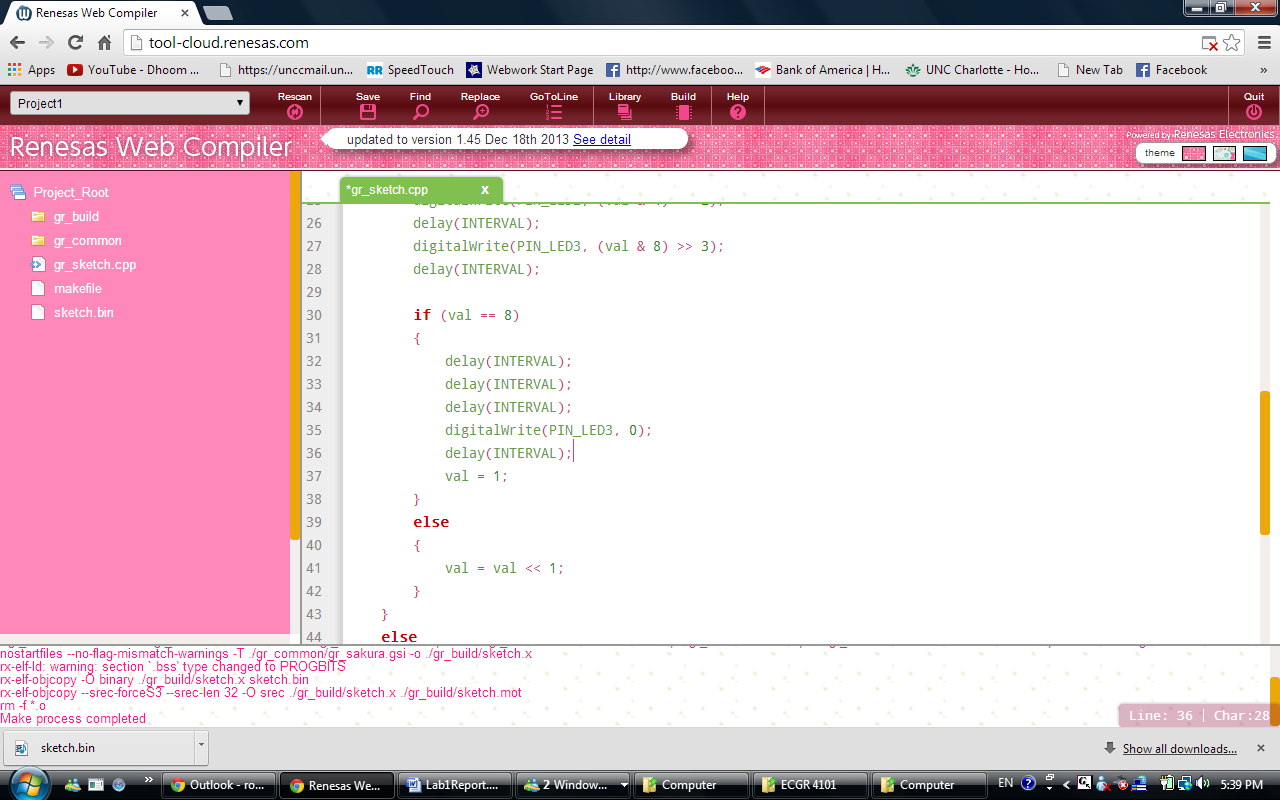
1. Each pin location is masked based on a binary representation from. Their location is compared against an indicator which will tell the LED Pin to turn on or off. LED0 is masked using. When “val” is equivalent to the indicator of ‘1’ then the location at will turn on. If “val” is holding a ‘2’ then the first location of would be off, but the next location of would be on. Between each LED comparison there is a delay inserted. This delay is used to allow the operator to see the changes in the LED states. Without this delay the LED’s could switch so quickly that some changes would be impossible to view.



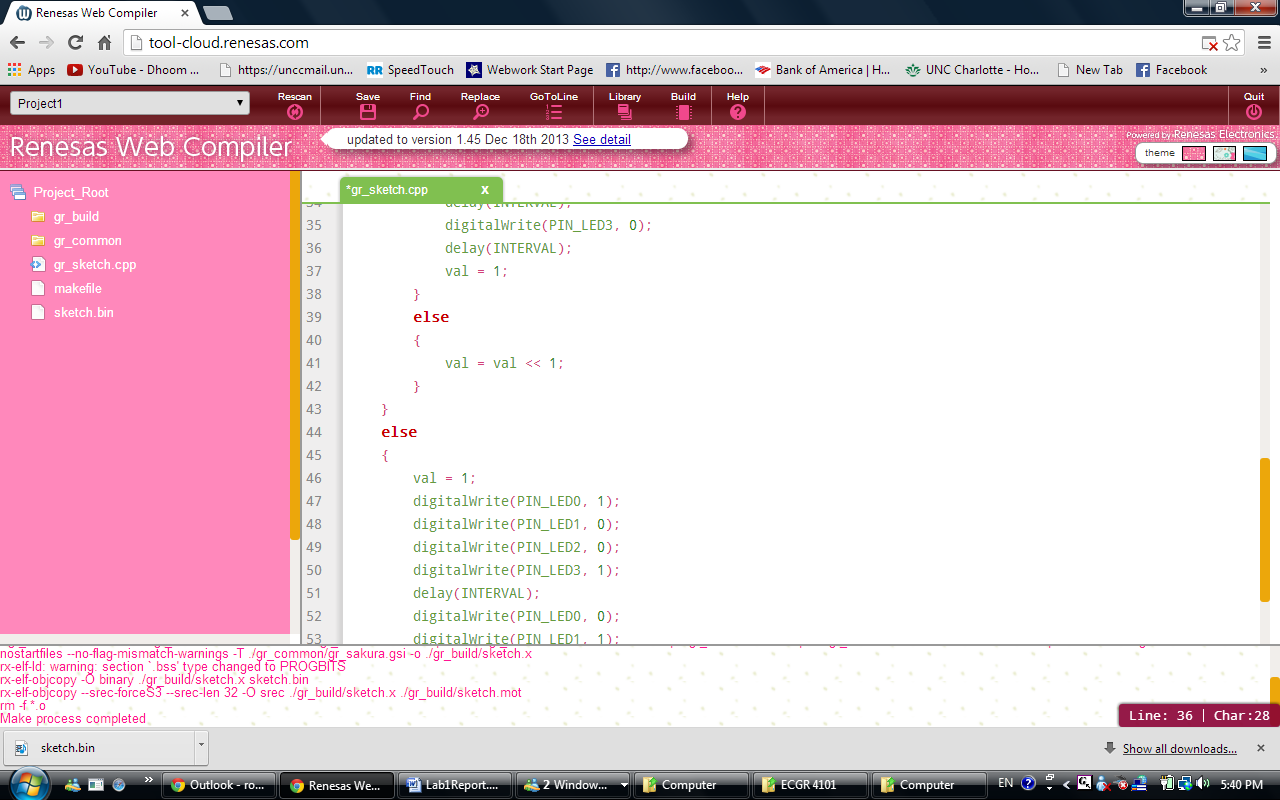
1. These lines will set the delay for 3 intervals in order for the viewer to see the reset of the counter. With no delays it takes no time to reset and the counting process resets



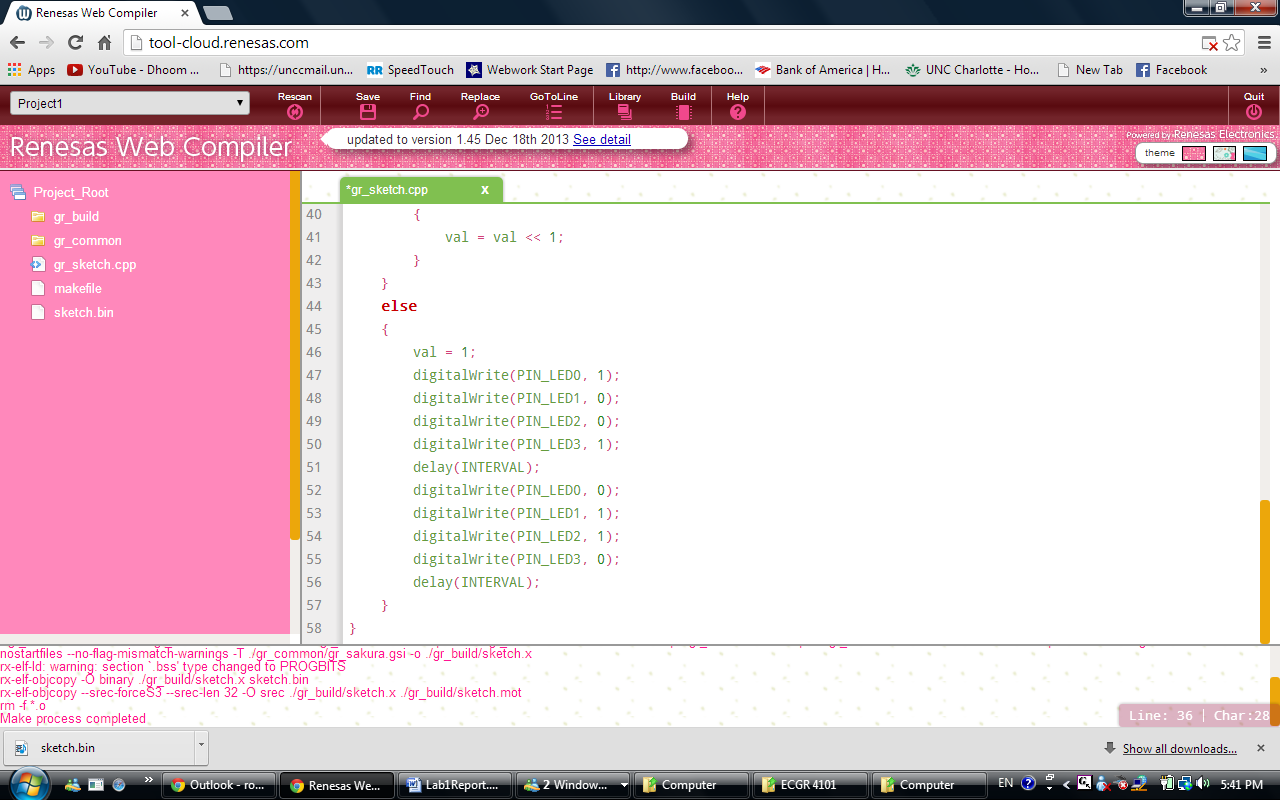
1. Writes the 4th LED to 0 so that LED3 and LED0 are not on at the same time
2. Last delay between reset of LED's; val = 1 will reset the counter to 1 to start at LED0



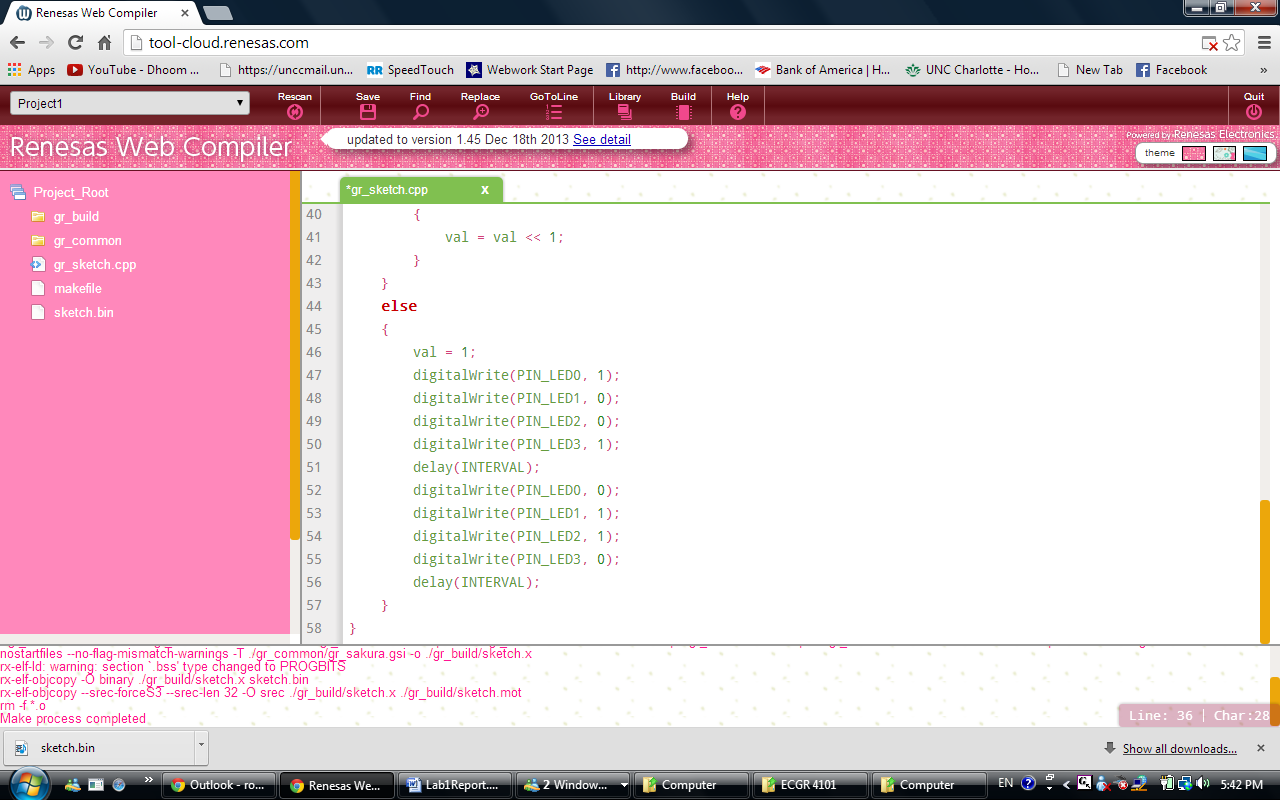
1. resets the value incrementer to 1



1. when the switch2 is not pressed; the LED configuration will be (LED0)(LED1)(LED2)(LED3)  with the first pattern being (ON)(OFF)(OFF)(ON) and there will be a delay for 200ms before continuing part of the loop



1. The Second Pattern will be (OFF)(ON)(ON)(OFF) and there will be a delay for 200 ms until going back to the beginning of the loop



**Observations:**

**Waveform when SW2 button has not pushed:**

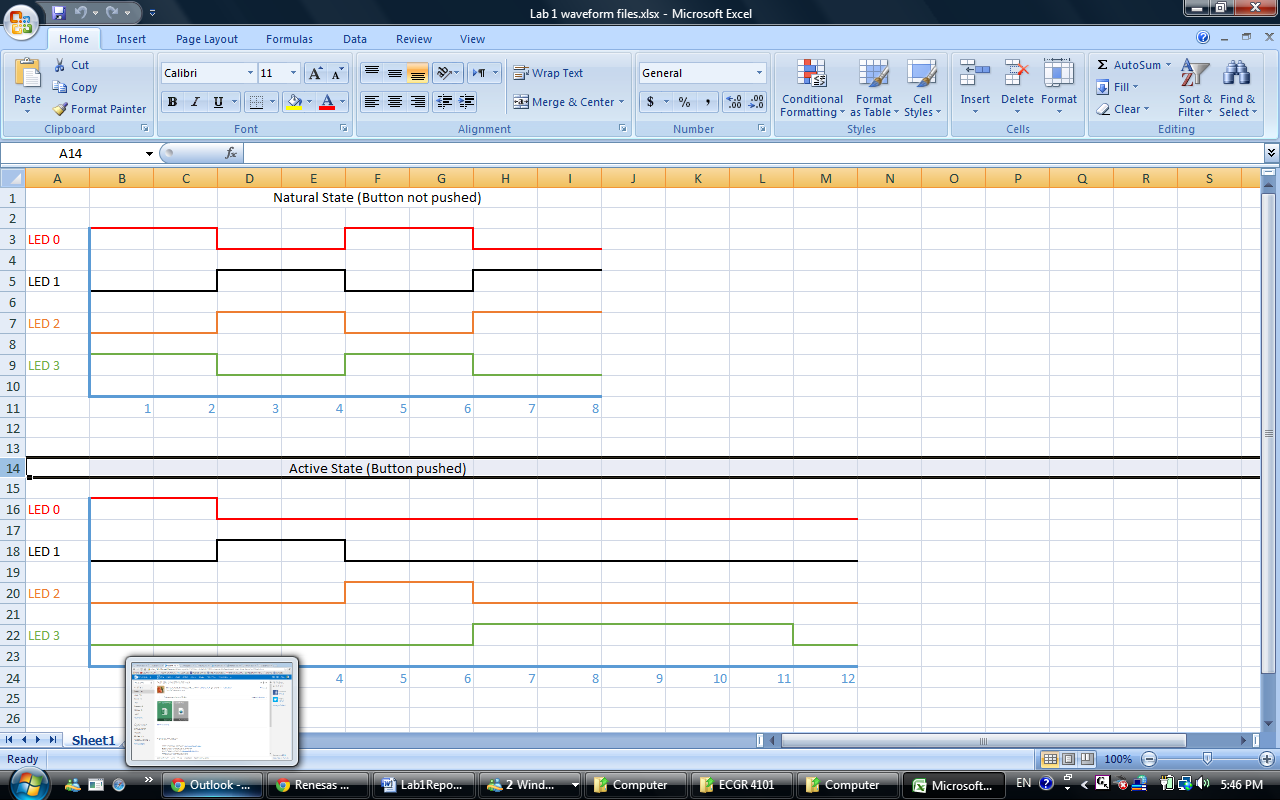


Figure . Natural State

**Waveform when SW2 button has been pushed:**

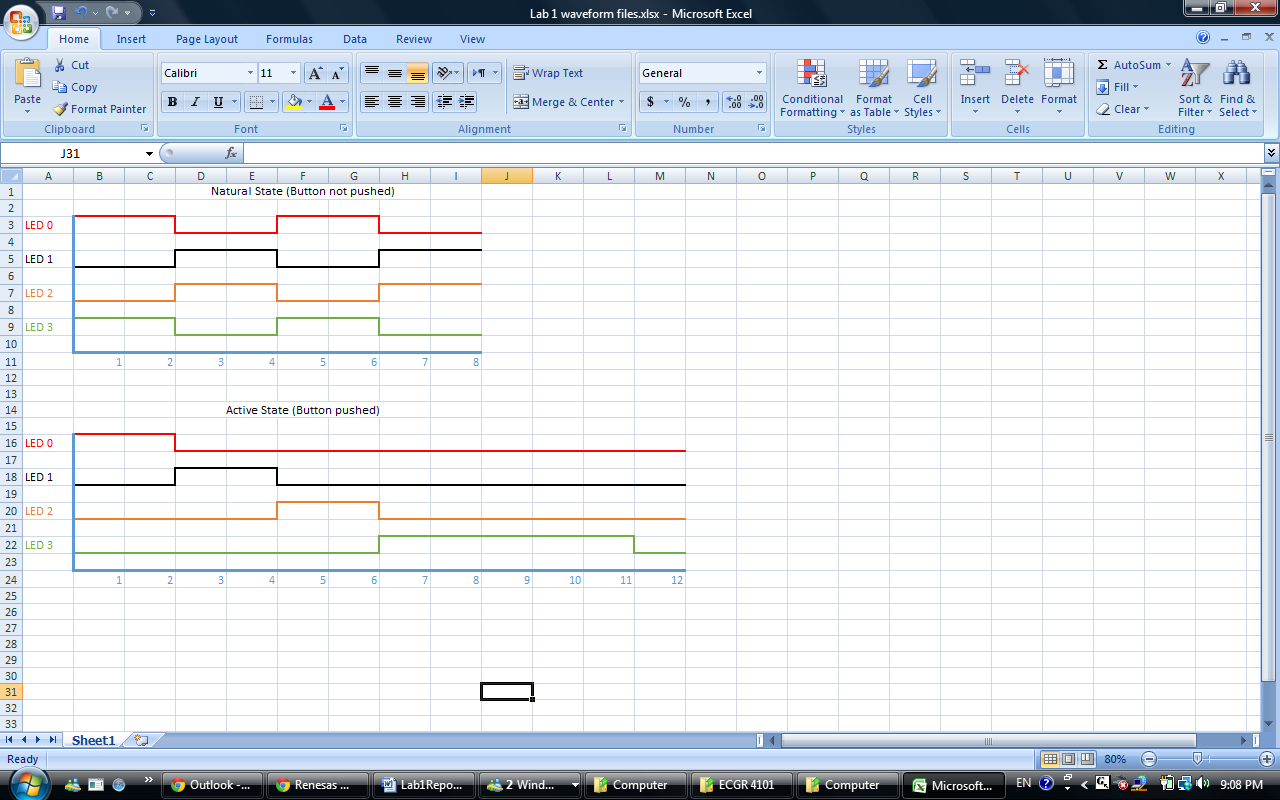


Figure . Button Pressed

**Conclusion:**

In this lab, we learned how to compile and run a basic C code by using Renesas Sakura board and the web compiler. The code itself taught the principles of delays, bit masking, and loops to control states. These basics will be useful in upcoming labs when trying to solve more complex problems. Overall this lab was fairly straightforward and resulted in no problems.