SWE 699/CS 695
Traffic signal Implementation
10/16/2022

Aastha Neupane (G01255142)
Gargi Sontakke (G01334018)
Preeti Bhattacharya (G01302375)
Uday Kumar Kamalapuram(G01340201)

Note: For the BeagleBone Black code, please install BBBiolib library in the system.
Library source code Link: https://github.com/ML-Cai/BBBIOlib
Reference Link to install:
https://community.element14.com/products/devtools/single-board-computers/next-genbeaglebone/b/blog/posts/beaglebone-black-bbb-and-pocketbeagle-i-o-gpio-spi-and-i2c-library-for-c-2019-edition

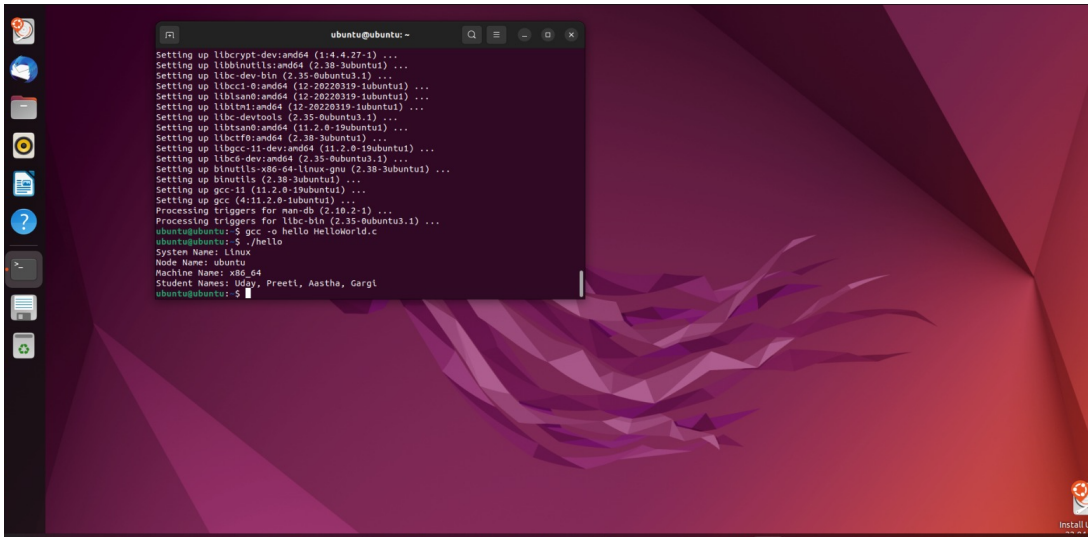
Please let us know if you face any issues in installing the library, Library file must be installed to run our BBB code.

1) **Code you used for testing on QEMU and the screenshot of QEMU output.**

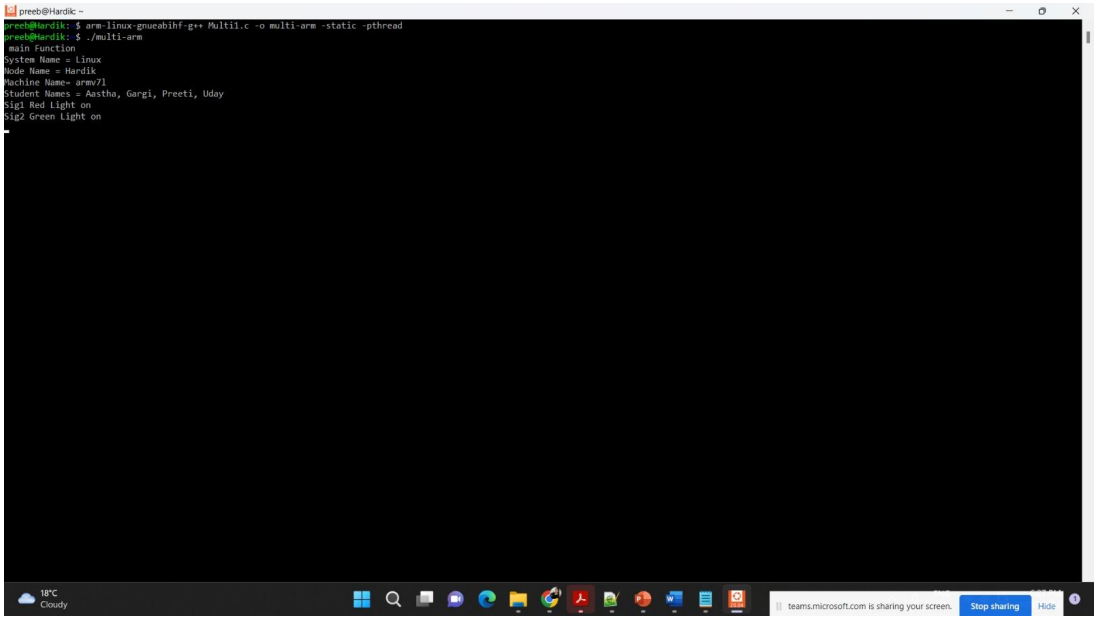
Ans: **Attached is a **multiTrafficQEMU.c** for source code

Terminal Screenshots:

utsname information:



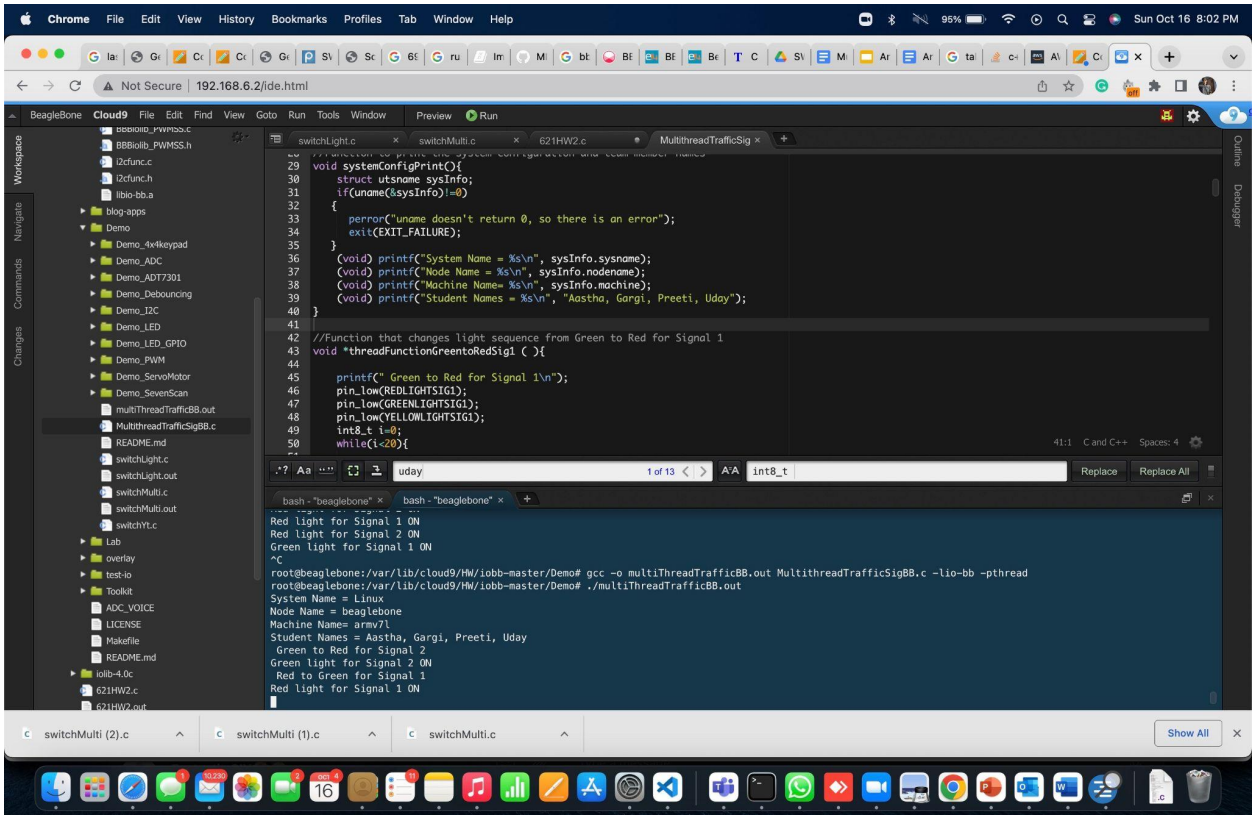
Testing on QEMU:



2) Code for BeagleBone

Ans: **Attached file **MultithreadTrafficSignalBB.c file

Terminal Screenshot:

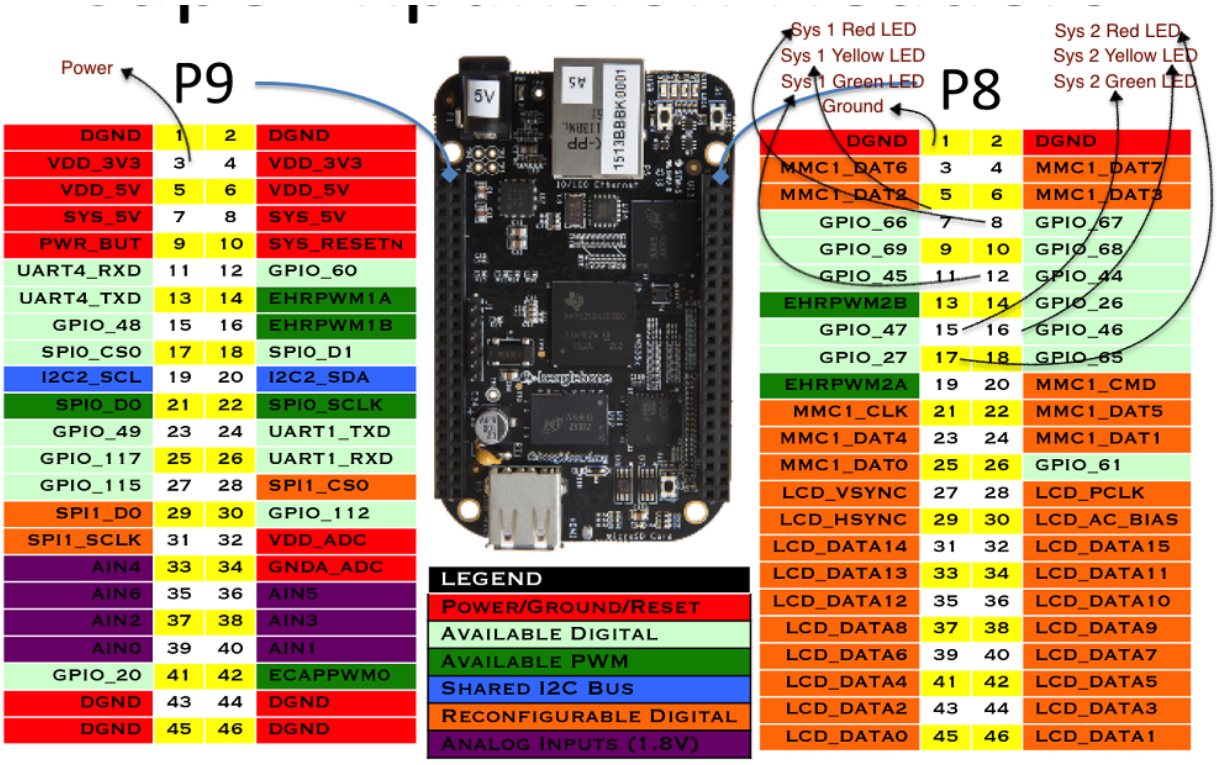


3) Description of your hardware design - this can be a schematic or simply a text document describing how the LEDs were connected and to which ports on the Beagle.

Ans:

The LEDs (2 of each: red, yellow, and green) pinned to a Breadboard were connected to the various ports/pins in the BeagleBone Black. A list and diagram showing where exactly the LEDs were connected on the BeagleBone Board are below:

- Red LED 1 connected to GPIO 67
- Yellow LED 1 connected to GPIO 66
- Green LED 1 connected to GPIO 44
- Red LED 2 connected to GPIO 27
- Yellow LED 2 connected to GPIO 46
- Green LED 2 connected to GPIO 47
- Connected Ground to DGND 1 on connector P8
- Connected Power to VDD_3v3 on connector P9

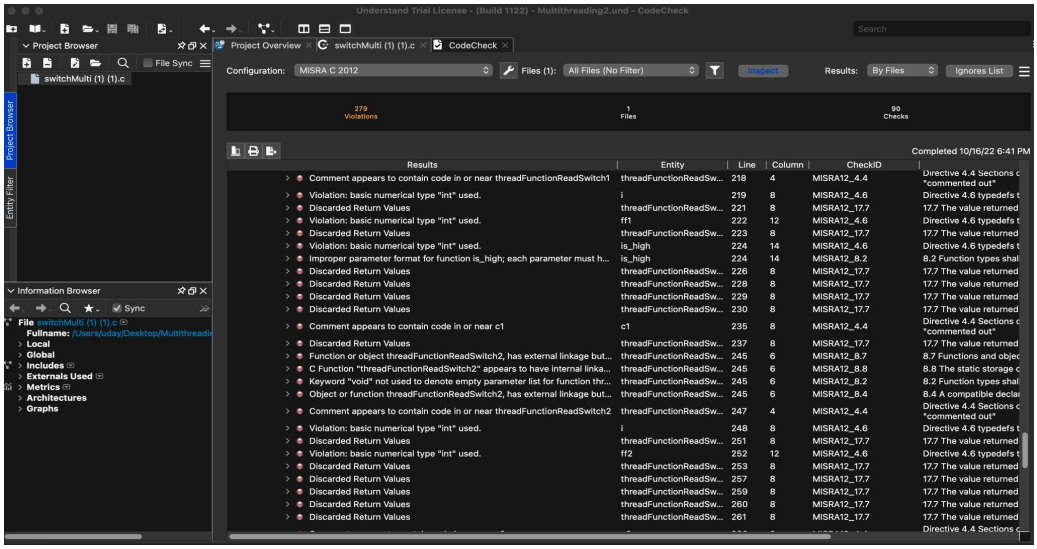


In the diagram above, refer to the green bordered area for the specific connection info

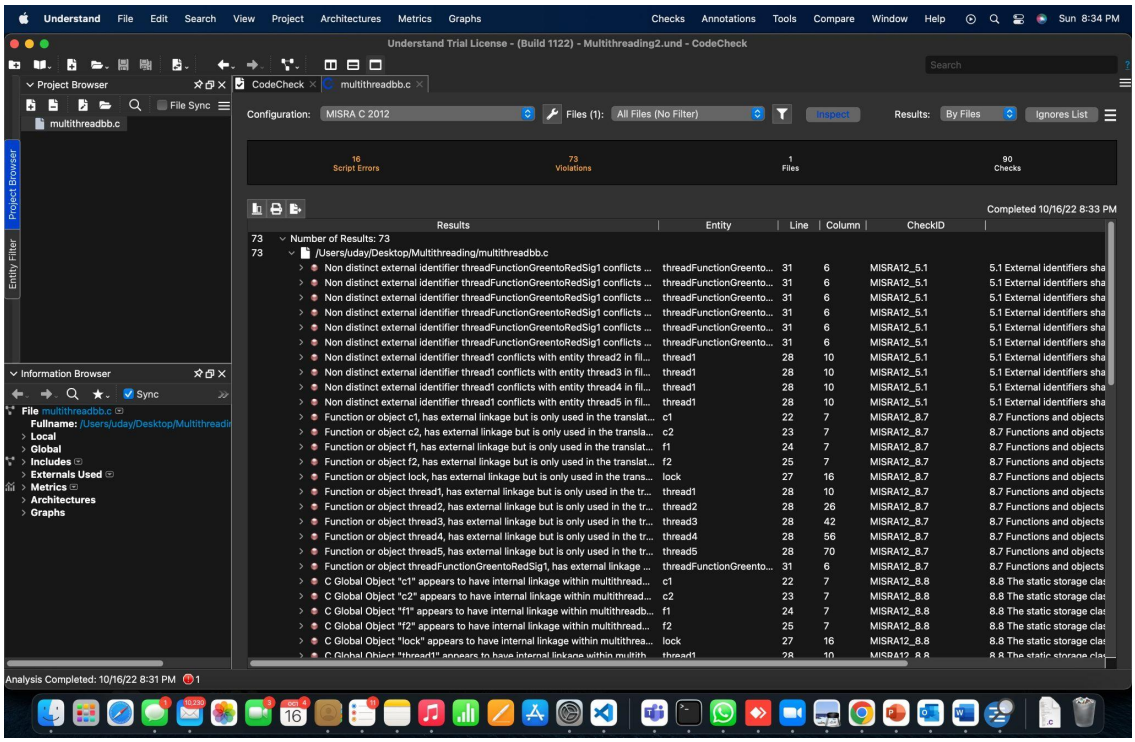
4) CodeCheck report from Understand on the Beagle code showing compliance with MISRA-C

a. **You do not need to get zero warnings/errors from CodeCheck - that's nearly impossible. However, I do want a reflection on the CodeCheck output. Why didn't you (or couldn't) address all the warnings?**

Ans: When we executed the first compliance check on the code, there were 279 compliance violations.



We were able to fix most of the violations, by researching them online and trying to make a few code changes. The violation count came down to 73 after the change.



We could not remove/fix the remaining errors because fixing those resulted in the code not working properly in physically implementing the traffic lights with BeagleBone Black.

We fixed a violation that was Directive 4.6 (Basic numerical type int used) which required typedefs that indicate size and signedness to be used. We fixed this violation by using ***int8_t*** (to provide an exact bitwidth/value range) instead of ***int***.

We fixed another violation of rule 8.2, “Function type will be in prototype form with named parameters” by adding **void** as the parameter type in the main function.

There were other violations that we chose to not address. The reason is that fixing these violations resulted in the program not executing properly on BeagleBone Black and thus not giving the expected external output of the traffic signal lights.

5) Video of your working lights and the wait sensor
Ans:
Please see the attached file for video demonstration of this assignment.