SWE 660 : SOFTWARE ENGINEERING FOR REAL-TIME EMBEDDED SYSTEMS
ASSIGNMENT : PROGRAMMING TRAFFIC SIGNAL

GROUP 9

G01409576 Shiva Nandh Reddy Mokilla
G01411756 Priyesh Rajesh Patel
G01389944 Charan Paladugu
G01394233 Anusha Jagadish

FILES INCLUDED

- traffic_signal_beaglebone.c : C code for Beaglebone
- traffic_signal_QEMU.c : C code for QEMU
- CodecheckResult_BeagleBone.pdf : MISRA report for Beaglebone
- CodecheckResult_QEMU.pdf : MISRA report for QEMU
- QEMU.png : Working screenshot of QEMU
- Recording_Assignment2.webm : Working video recording of Beaglebone
  Recording-20240227_225903.webm

PROJECT EXECUTION OVERVIEW

- Developed C code for the traffic signal system.
- Tested the code on QEMU emulator to ensure functionality.
- Established hardware connections on the BeagleBone.
- Conducted thorough testing of the system.
- Downloaded and utilized the Understand tool to generate a MISRA C report.
- Made necessary modifications to the code to comply with MISRA standards.
- Re-tested the modified code to ensure continued functionality and compliance.

MISRA STANDARDS DISREGARD RATIONALE

- Directive 4.6: Typedefs for Size and Signedness
  We chose to ignore this directive as using basic numerical types like int or short provides convenience and readability in code. While typedefs offer precise control, the readability and ease of use of basic types were deemed more suitable for this context.

- Directive 15.5: Single Point of Exit in Functions
  This directive was disregarded as enforcing it could lead to unnecessarily complicated code, potentially making it harder to understand. Prioritizing code cleanliness and ease of comprehension outweighed strict adherence to this guideline.

- Directive 21.6: Standard Library I/O Functions
  We decided to overlook this directive because standard library I/O functions like printf and scanf are widely supported, well-documented, and familiar to many developers. Utilizing them simplifies code development and readability without significant drawbacks.

- Directive 8.2: Function Declaration and Definition Separation
  We ignored this directive in situations where combining the declaration and definition of functions simplifies the code without compromising clarity. In smaller programs or specific contexts, this approach can enhance code readability without introducing significant drawbacks.