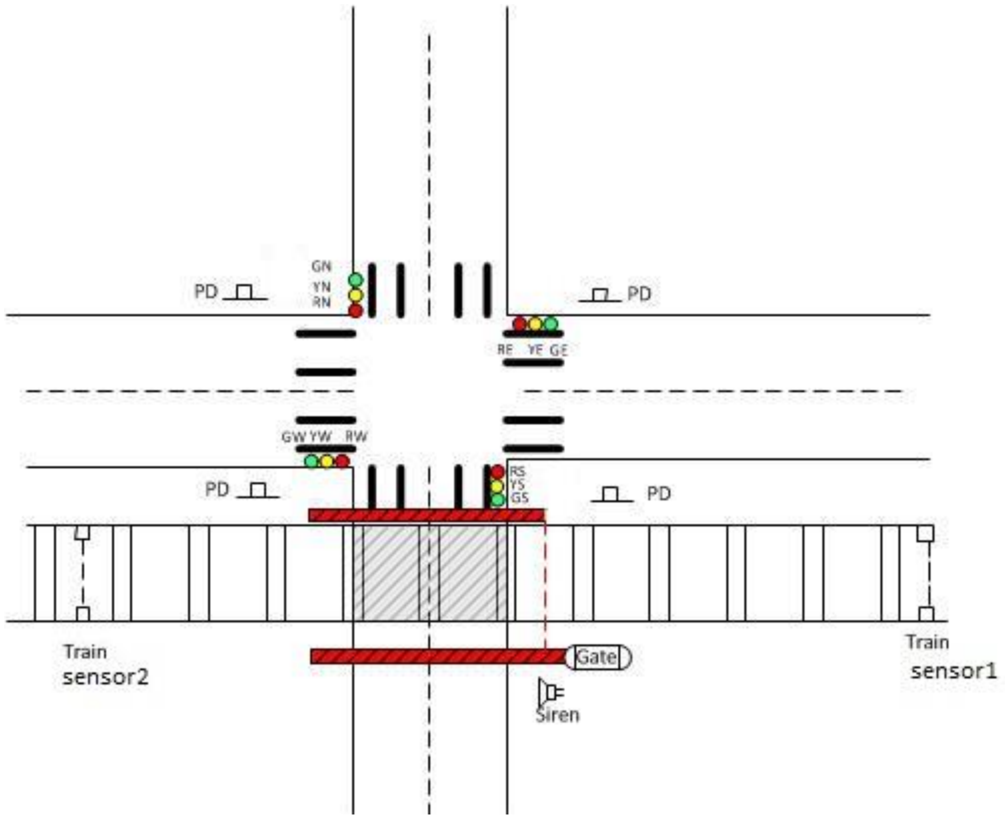
Multi-junction traffic light controller

The goal of the class project is to give you experience in Real-Time Embedded System Design. The design experience will utilize one of the prominent processor technologies, namely, the application specific instruction set processors, or microcontrollers. The TivaC board will be the hardware platform for the project implementation and the FreeRTOS will be used to manage the concurrent tasks.

This course project is to design a Traffic Light controller with four road junctions of different priorities, with parallel railroad crossing. The railroad goes east/west and has the highest priority. The north/south road junctions have higher priority than the east/west roads. Single lane railroad crossing has a gate, which is vertical by default, and closes before train approaches. Once the gate is about to close, all four red lights go flashing, (and with a bonus, a siren sounds is turned on) until the train passes. Pedestrians can also provide requests to cross, at any of the four corners. System can operate at various modes as specified later. See layout below of the crossing:



Each road junction has its timer for the Go (green light), followed by Stop time (sum of the Go times for the other junctions), when other directions can Go. As stop time expires, Green light of the corresponding junction goes green, and all other opposite directions must be red. A green light should be followed by a yellow light before it turns red, but this feature will be removed for simplicity. Also, for simplicity, assume that North and South Go together, as well as East and West Go together. Algorithm for calculating the order in which the Red lights have to glow is decided by the priority of the corresponding junction.

Operating modes:

Normal mode:

This is the normal mode. As spefied in the simplification features, either the North/South Go or the East/West Go, with parametric times given at compile time: tgn=tgs, tge=tgw in this order. For example, if these values are 5, 5, 2.5, 2.5, then the Go times for the North/South will be 5 time units, and stop for 2.5 time units (seconds, minutes). Thus, the priority is higher for the North/South directions.

Pedestrian crossing/Hazard mode:

This is an as needed mode. Four PD buttons are provided at each of the four corners, to provide a single signal. When this signal is asserted, the current junction that is going waits for its remaining time, then all stop lights should be turned red for time equal to compile time parameter: tcross. After that time, priority mode should be resumed. Same action can be done when an emergency vehicle approaches. All flows should be stopped.

Train approaching:

This is the highest priority mode. Two train approaching sensors (simulated by an on/off switch), are placed at either side of the rail road to provide a single signal 'Train'. If this signal is active, the current junction that is actually going should stop, then all stop lights should be flashing red. Then the train gate should close (simulated by turning on a RED light, or for the bonus option use a servo motor and attach a gate bar to move from vertical to horizontal position, also for a second bonus turn on a siren). The distance between the two train sensors should be longer than the longest train. So, when the second train sensor is triggered it should be assumed safe to open the gate and resume the normal mode.

Desired features/assumptions:

- Two contact switches will be used to simulate the approach of train, and that the train has cleared the crossing, in either order. For example, if the train is approaching from west to east, then sensor 2 should be on first (close the gate), then the gate should open when sensor 1 is on. The time given to cross any train should not be less than tsafety (for safety consideration).
- A single servo motor controls could be used (in bonus activity) to close/open the railroad crossing gate. Otherwise, a RED LED should be used.
- Indicate status information on the debug console of the IDE.
- Use a single push button to simulate PD (pedestrian request to cross) on any of the four corners of the intersection (or hard wire all four to give a single input).
- Use a push button for reset. On reset, assume the controller starts in normal mode, and put north/south to Go, and East/west to Stop, and start all timers.
- You may use PWM or A/D to sound the siren.
- FreeRTOS has to be used in priority mode to schedule all tasks.

Test parameters:

fref=1Hz (to blink the red lights)

tgn=tgs=5 s

tge=tgw=2.5 s

tcross=10 s

tsafety=30 s

Demos: Make your own prototype. Record a video demonstrating the working of your project and send to the TA's email (mostafa.a.arafa91@asu.edu.eg).

Apply best design models to represent your design. Make sure the code fits the limitations of the IDE used to program the TivaC board. Use µVision IDE, CodeComposer or any other IDE to test/debug/flash the code into the TivaC board

Schedule:

02/26/2020: Project Introduction, and team formation. Team leader: send the TA an email including names of your team members.

End March/2020: Full team submission, as detailed below.

End March/2020: Individual report submission, as detailed below.

General Requirements and Constraints

- Work in groups of **three to five** students each.
- You select your group members. Each team should elect a team leader. Notify the TA via email with team members and team leader, no later than a week after project Introduction.
- Each member must design part of the project and must write his/her **OWN part of** the report. Each member must **clearly** show his/her individual contribution to the project.
- Essential to help each other in the same team, but not across teams!
- All diagrams must be done on the computer; Use Program State Machine Model to layout your system software components and their dependencies.

The Full team submission report should be divided to two main parts:

System Design Part

The system design part should have the following contents in order:

- Title Page with project title, ASU class number and title, date of submission, team name, and members.
- Project Specifications and Description, expanded and clarified from these given specs.
- Design Choices, including any choices you made that were not given in the specification or any functionality you will expand.
- Detailed block diagram (graphic) and diagram descriptions.
- Team member responsibilities for the project.
- Plan of timeline: what is expected to get done, and at when. Make a proposed timeline.

Technical Details Part

This part includes following contents in order:

- Design details, operation manual. What should we do to test your prototype? This can be documenting what you present in your demonstration video.
- Code listings. What functions you used, and what modifications you made. What new functions you designed.
- **Lessons learned (important)**
- Problems faced (if any)

Each team will email to the TA a **zipped file** which contains all electronic documents related to this project (compressed software project files, final report document) plus a video URL of your demonstration. The video should have audio narration (explaining all test cases of your system).

In addition, each **team member** will submit a separate short report to the TA's email which includes following contents in order:

- Title Page (team name, your name, class number and name, submission date)

- Your **individual contribution**, Be specific.

- Evaluate your team members' work.

- Provide any suggestions or comments on the project. There is always something that can be done better.

Grade = system design part (20%) + technical details part (60%) + report organization, and clarity (10%) + teamwork (10%).

If some team members are left not performing, this will usually affect everyone else's performance. So, try to resolve such problems yourselves, early on, and don't ignore them. But if needed, you will have to report it to TAs and they will report to the course professor.

Deadline will be enforced: We will deduct 5 points per day past deadline.

With my best wishes,

Prof. Dr. Omer Alkelany