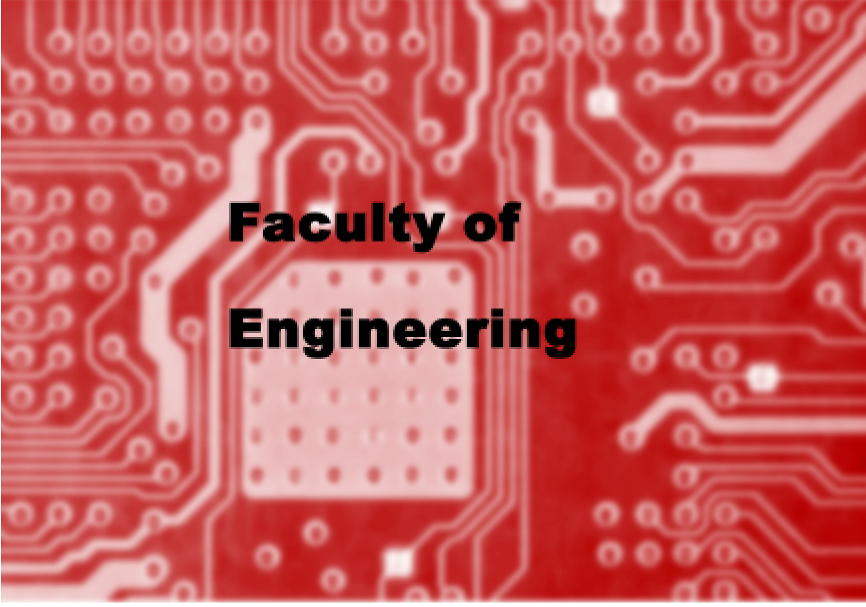
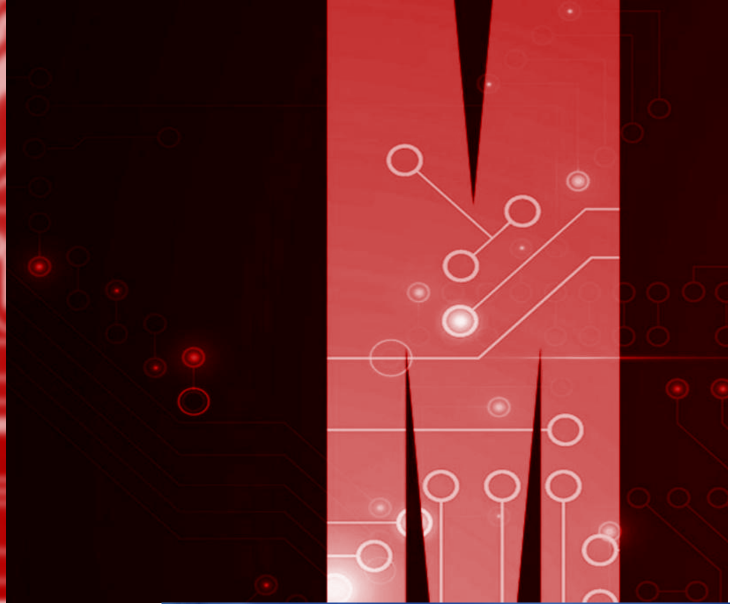
Faculty of Engineering

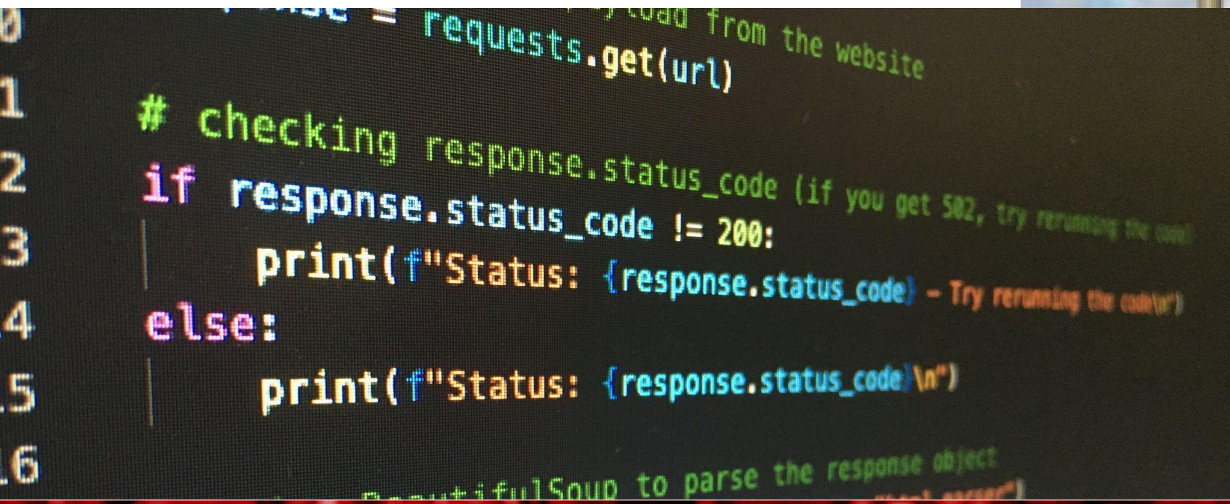


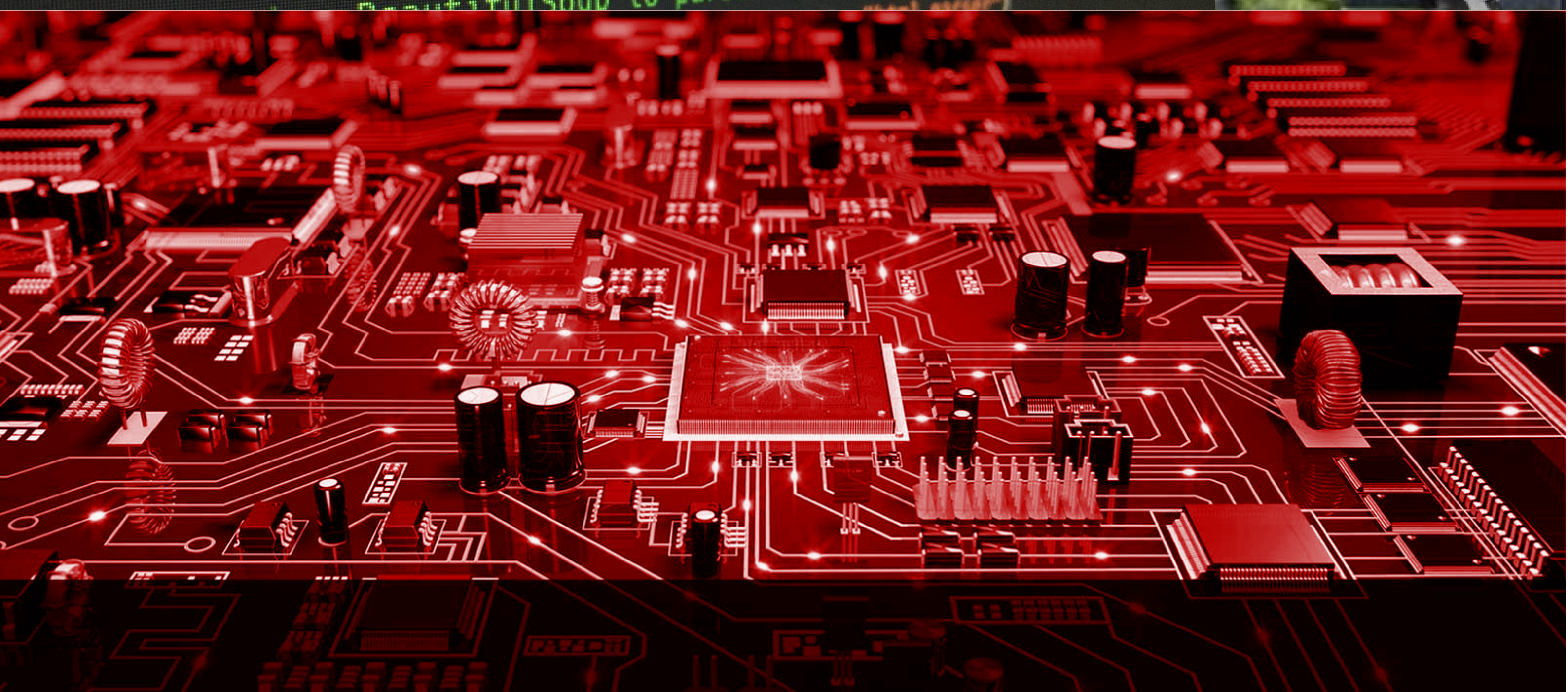


ENG1013: Engineering smart systems

Traffic Control System Project

Project Specifications 2025





1 Preface

“Smart systems” are critical to the future of engineering, as they address economic, environmental, and societal issues. They are utilised across many industries, including energy supply, healthcare, agriculture, manufacturing, security, entertainment, and logistics. Some examples of smart systems are automatic lighting and climate control in buildings, responsive traffic light controllers, liquid level monitoring in agricultural and industrial settings, intruder alarm systems, driver assistance systems in automotive vehicles, and personal healthcare devices. In smart systems, sensors and timers are often used to detect certain events. Pre-programmed software is then used to perform analysis. This analysis can be used to make decisions and perform actions when combined with historical data.

In this project, you will work with your team to apply your software, electrical, electronic and mechatronics engineering knowledge learned throughout ENG1013 to build a **responsible traffic control system.** This type of traffic control system will control vehicular and pedestrian traffic and is similar to those found at road intersections and pedestrian crossings to help control and manage traffic flow. These systems can be activated based on timing devices, the detection of vehicles, or the press of a button.

In the course of working on the project, it is completely normal for engineering design tasks to feel overwhelming or confusing at the beginning. You're stepping into a process that involves solving complex problems, often without all the answers in front of you. This is an exciting challenge, and here's an important tip to help you succeed: **ask questions—lots of them**.

When you're working on a design project, think of yourself as a detective. Your "client" (whether it's your professor, a real-world client, or even a project brief) has a vision for how the system should work, but they might not give you every detail upfront. Part of your job is to ensure that you’ve understood (and obtained!) all the critical information so that you (and your team) can build an engineering solution that correctly meets the requirements.

Here are a few supportive reminders for approaching this:

- **Confusion is part of the process**: Feeling uncertain is a sign that you're in the middle of learning something new and challenging. Instead of being frustrated, use that feeling to start digging deeper.
- **Questions lead to clarity**: If something doesn't make sense, or if you're not sure about a requirement, ask! There’s no such thing as a “silly question” in engineering design. Often, asking the right question can save you hours of effort and guide your team toward better solutions.
- **It's a collaborative effort**: Engineering design isn’t done in isolation. Asking questions isn’t just for you—it also helps your client clarify what they want you to build, which might not be obvious at all.
- **Learn to listen and observe**: Sometimes, answers you need aren’t just in the words a client says—they’re in how they say it, the constraints they emphasize, or even the problems they didn’t mention.

Remember, every great engineer started where you are right now. It’s okay to feel confused—embrace it as an opportunity to learn. By asking thoughtful questions, you’ll build not only your understanding but also your confidence as you tackle these exciting design challenges. You’ve got this!

2 Project Resources

2.1 Index

Here's a handy list of activities in the lab manual that link to various components of the project. You can find component videos and datasheets on Moodle, under Learning > Project Resources > Guides and Datasheets.

2.2 Standards

You must follow these "standards" when formatting your code and producing circuit diagrams. Standards are important to ensure that your designs can be understood and reproduced by other engineers.

Circuit Diagram Standard Template (Draw.io) (Click > Open with "Draw.io")

Circuit Components Library (Draw.io) (Right Click + save to your computer > Go to "Draw.io" > File > Open Library From > Device > navigate to the file you downloaded)

Coding Standards Document

Circuit Standards Document

2.3 Help and Support

There are many avenues available for you and your team to get help with the project.

- You can ask your manager questions during the weekly practical class.
- You can ask questions on the EdStem forum. If the question is specific to assessable code (eg debugging) please submit it as a private query, as per the forum guidelines.

Remember that no question is a stupid question. Your manager (and all the other managers) want you to succeed and are always happy to guide you.

Students also have access to the Electronics lab 5205 during working hours and Digital Storage Oscilloscopes available in the location for use. Should you want to take wire strippers home to use, each team may loan a pair from us.

Please note that support for any milestone will terminate one working day at 6pm prior to the deadline, in the week that it is due, with the exception of Milestone 2 which will terminate at 6pm on the day of the deadline.

3 Project Restrictions

Your team has been asked to design, build, and demonstrate a simplified traffic control system in this project. Your system will consist of multiple subsystems, and the client has split these subsystems into groups of *required* features, *general* features and *integration* features. There is also an additional *complete integration* point value assigned to the final Milestone of the project. Most subsystems will contain multiple *general* and *integration* features which your team can choose to implement for the final Milestone, which you can read about in the milestone deliverables section, and in the system features section below.

The project is graded out of 260 points, with the final point value being scaled based on your individual contributions as a team member, and your final viva interview. Your mark is scaled according to the Weighted Scaling Factor to form 26% of your total mark for ENG1013. This project is not part of any hurdle requirement. You can find rubric feedback under the relevant link in Moodle under ‘Learning > Project Resources > Results’ which shows your team’s performance in each milestone against each assessed item. Your demonstrator will return feedback to you when marking during the milestone demonstration in class.

3.1 Hardware Requirement

You are permitted to use any and all of the hardware components provided in **all** of your team’s supplied kits (i.e. all 5 combined), plus single core wire and additional breadboards. However, you may use only **ONE** Arduino to attain maximum points for a fully integrated system. If your team isn’t aiming to attain points from the ‘integration features’, you can use multiple Arduinos - one for each subsystem - note that only one Arduino can be ‘active’ and running at once. You may not use any other hardware components. Read more in the dedicated Integration section below.

3.2 Software Requirements

You may develop your software in any Integrated Development Environment (IDE); however, support will only be provided by demonstrators in VSCode. You may only use the following additional packages within Python 3.10.x:

- Pymata4
- Time
- Math
- Matplotlib
- Random

The following packages are permitted, but demonstrators will not provide support:

- NumPy
- CSV

All other packages, Classes and assert statements are not to be used. If found, they will be commented out for marking.

4 System Overview

This project is based on the Blackwall Tunnel (Southern Approach) which exists in the UK. We have simplified the actual system in reality to comprise of the A102, the freeway leading up to the tunnel entrance, and Tunnel Ave, the arterial road running parallel to A102 near the tunnel entrance. The main emphasis of the system is on traffic control and to prevent or minimize overheight vehicle collisions at the tunnel entrance, which only can accommodate up to 4.0m high vehicles. The system provides a overheight detection system on approach, a final overheight vehicle exit and a final overheight vehicle detection system just prior to the tunnel entrance. There are also systems in place to take control of the traffic should power fail in the area, which is run off backup batteries. A model of the overall system is provided below. You can find the feature and subsystem specifications under System Features which describes the system behaviour.

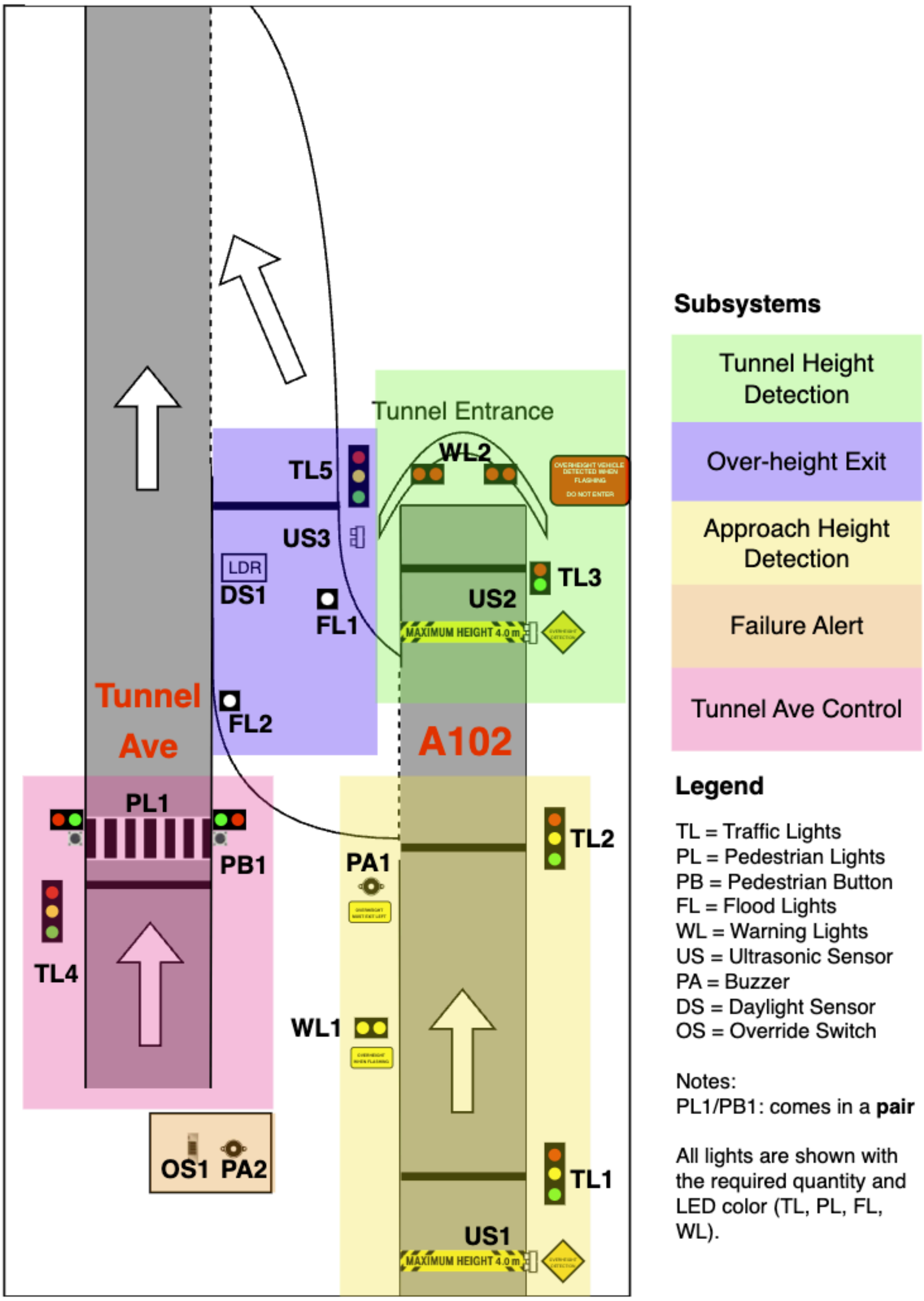


Figure I: Simplified Model of Traffic System

5 Deliverables

The project is split into three milestones, where the first milestone is a design/plan stage, and you will be building an incremental product for submission over milestone 2 and milestone 3.

You will earn **points** over the course of this project, which will translate into marks after the completion of the full project (and completion of the viva interview). These marks will be individually scaled based on a Weighted Scaling Factor using your third ITP Metrics Peer Assessment Score (PAF3) at 50% weighting, and your Viva Interview Score at 50% weighting.

There are three project milestones:

1. Delivery of a design document (50 points) which includes
    a. Hardware system interaction block diagram showing all interaction between hardware components as blocks
    b. High level software logical flowcharts showing all interaction between components, one flowchart for each subsystem
    c. Planned timeline of work (gantt chart)
    d. Communications plan
    e. Conflict resolution plan
2. Initial system prototype submission and demonstration (70 points)
    a. Circuit diagrams (one circuit per function / per page as a .pdf file) for all components being demonstrated
    b. Software code (.py python files) for all work being marked
    c. Physical circuit (on demonstration day)
3. Full system submission, demonstration and viva (oral interview) (140 points)
    a. Circuit diagrams (one circuit per function / per page as a .pdf file) for all components being demonstrated
    b. Software code (.py python files) for all work being marked
    c. Physical circuit (on demonstration day)

For Milestones 2, and 3, your prototype **must be demonstrated in person** during your lab session **to achieve the maximum points available** (100%). However, suppose some part (or whole) of your prototype is not functional on the day. In this case, you may ask for a **video your team submitted as part of your milestone on Moodle** to be marked instead for **66%** of the points available. If you have not submitted a video, your team may **explain** how your system should work for **33%** of the available points.

Individual bonus points

Knowledge gained from completing the Practical Learning Competency tasks will help you greatly throughout the project. If you complete these **within the first three weeks of the assessment offering**, you will be awarded 2% bonus marks for each, to be applied individually to your final project mark (capped at 26%).

5.1 Milestone 1

This milestone is due at the end of Week 4 on Friday at 11:55pm.

5.1.1 Milestone Deliverables

System Design Document for the initial prototype (Milestone 2). This document includes the following sections (in this order). You are recommended to use the provided template for your submission that explains the following items. Further examples for the document are available on Moodle.

- System Interaction Block Diagram showing all hardware interactions for milestone two.
- High level flowcharts for all software features (with hardware interactions) for milestone two.
- Proposed Timeline for work showing scheduled time to work on milestone two features.
- Communications plan showing your team's primary, secondary and backup communications methods, along with the expected response time in varying scenarios.
- Conflict resolution plan showing your team's resolution and escalation process for disagreements and conflicts, including when work is not done to the agreed standard, or if work is not done on time.

5.1.2 Milestone Weighting

This milestone is marked out of 50 points. Points value breakdown as follows:

- System Interaction Block Diagram: 12 points
  Marking criteria:
  - Includes all hardware components required for M2
  - Components clearly belong to a specific subsystem on the diagram
  - Hardware components are logically linked to each other appropriately
- High Level Flowcharts: 20 points
  Marking criteria:
  - Items in flowchart are at a high level (one flowchart per subsystem)
  - Proposed flow of logic is sensible
  - Includes all relevant software components required for M2
- Proposed Timeline: 8 points
  Marking criteria:
  - Timeline shows proposed progression of work for all features required in milestone 2
  - Timeline shows work breakdown by member (assigns work to each member fairly)
- Communications Plan: 5 points
  Marking criteria:
  - Plan shows primary, secondary and backup comms methods
  - Plan includes expected response times in more than one sensible scenario
- Conflict Resolution Plan: 5 points
  Marking criteria:
  - Plan shows team resolution/escalation plan for resolving conflicts and disagreements
  - Plan shows team plan for handling low quality or incomplete / late work

Please note marks for the milestone will not be finalised until completion of the full project, after application of the Viva and PAF scaling (see section 4.2.6.3 and 4.3 for more details on scaling).

5.1.3 Submission Format

Submit your document as a single PDF file, and name it TeamNNN-M1.pdf, replacing NNN with your team code, i.e. if your team is MS02, your filename should be TeamMS02-M1.pdf

You should ensure that the contents of the diagram are readable. The page for the diagram can be as large as A3 Landscape, with the rest of the pages being A4 Portrait.

5.1.4 Deductions

Category	Description	Deduction
Overall Submission	Incorrect submission format (filetype not PDF, structure) or left instructions in template (didn't delete blue text)	-3
System interaction block diagram quality	System Interaction Block diagram for hardware is of poor quality	-3
High level flowchart quality	High level flowchart for software is of poor quality, lacking clarity, completeness, or accuracy in its representation. (includes readability / diagram flowing across pages in a jigsaw puzzle, wrong flowchart shapes, or have parallel logic)	-3
Proposed Timeline quality	Proposed Timeline incomplete or poor quality, lacking thoroughness, clear milestones, or a realistic allocation of tasks and deadlines.	-3
Comms Plan quality	Plan is incomplete or of poor quality (brief, isn't detailed)	-2
Conflict Plan quality	Plan is incomplete or of poor quality (brief, isn't detailed)	-2

5.2 Milestone 2

Milestone 2 is due at the end of Week 7 on Thursday at 11:55pm, with demonstration and marking done during class in Week 8.

5.2.1 Milestone Deliverables

Your team must demonstrate a functioning integrated initial prototype (70 points) that consists of the following:

- Code implementing required features (in blue) of following subsystems (70 points):
  - Subsystem 1 (Approach Height Detection Subsystem)
  - Subsystem 2 (Tunnel Ave Control Subsystem)
  - Subsystem 3 (Over-height Exit Subsystem)
  - Subsystem 4 (Tunnel Height Detection Subsystem)
- Circuit Diagrams for required features of the subsystems above. You will not be allowed to demonstrate circuits for which there are no circuit diagrams submitted.

Your team must demonstrate the built system's functionality to your demonstrator in your Week 8 class.

5.2.2 Milestone Weighting

This milestone is marked out of 70 points.

Marking Rubric for features:

- Functional and working without errors: 100% of points awarded
- Contains errors but can demonstrate was working in submitted video: 66% of points awarded
- Contains errors but can explain how to fix: 33% of points awarded
- Required or selected feature not attempted: 33% of points deducted
- Attempted but does not work, cannot explain: 0% of points awarded

Please note marks for the milestone will not be finalised until completion of the full project, after application of the Viva and PAF scaling (see section 4.2.6.3 and 4.3 for more details on scaling).

5.2.3 Submission Format

- **Circuit Diagrams:** The circuit diagrams must be in PDF format (.pdf), in a single file containing one circuit per page, and can be landscape A4 or portrait A4. You should have one circuit for each logical component. It is strongly recommended that you complete the circuit diagrams early so that your manager can review and give you feedback on them.
- **Code:** The code submitted must be in python format (.py), and can be split across multiple files if needed. Note that for integration marks, your code must execute from one single starting point (i.e. running the main.py file will start the entire system). In addition, you must have attempted all features.
- **Backup Videos:** Teams may choose to include a short video in .mp4 format (under 500MB) demonstrating the prototype functionality being assessed in their moodle submission. This will serve as a backup for assessment if the demonstration during class fails. If the file is too big, upload the video to your google drive and **include the link to the specific videos** in your submission as a text file.
- **Submission naming format:** Put your project code (.py files) and circuit diagram (single .pdf file) in a folder named TeamNNN, replacing NNN with the code for your team. Then ZIP the folder up and name it TeamNNN-M2.zip, replacing NNN with the code for your team, i.e. if your team is MS02, your filename should be TeamMS02-M2.zip (max zip file size 500MB).

5.2.4 Testing

During your scheduled Practical Class in Week 8, we will test the system you submitted. Your team will need to **arrive at class with the circuits required already built.** (i.e. no time will be given to you in class to build them). **If you are late or absent, your team might not get marked**, so make sure you arrive on time.

5.2.5 Deductions

A breakdown of all key deductions can be seen in the table below.. Please discuss any implementation requirements with your manager if you have any questions.

Category	Description	Deduction
Overall	No circuit diagram submitted for a feature.	Affected feature will not be assessed (no attempt given)
Overall	Other hardware components besides the allowed hardware components are used (items not provided in your project kit).	The project will be marked once the offending hardware components are removed - they are not to be replaced.
Overall	Non permitted packages, methods and techniques used.	Offending code to be commented out.
Circuits	Circuit Diagram does not match physical circuit submitted	Late submission penalty of two days (10%) or cap of feature mark to 33%
General Code Quality	● Console print statements are unclear, have insufficient information, or are poorly formatted. ● Console alerts are at an inappropriate rate (i.e. spams the console / prints multiple times per second). If code is used to clear the console, this deduction will automatically apply. ● Code crashes, does not exit cleanly, has an unhandled exception, uses unspecific exception handling, or exits with an error message.	-5
Physical Circuit Quality	● Physical circuit doesn't follow wire colour conventions, ● Physical circuit doesn't use power/ground rails appropriately ● Physical circuit uses spaghetti wiring (except for wires to the Arduino).	-5
Circuit Diagram Quality	● Circuit Diagram doesn't meet required standards ● Circuit Diagram components are incorrectly oriented. ● Circuit Diagram is correctly labelled with unique identifiers and values. ● Circuit Diagram uses wrong component shapes (must use shapes from the library provided).	-5
Coding Quality	● Code doesn't meet coding standards ● TODOs left in code ● Variables not in lowerCamelCase ● Functions not in snake_case ● Indentation not 4sp = 1 tab ● Function header documentation in wrong location, format, incomplete, missing or not present ● File header documentation in wrong format, incomplete, missing or not present	-5

5.3 Milestone 3

Milestone 3 is due at the end of Week 11 on Friday at 11:55pm, with demonstration and marking done during class in Week 12.

5.3.1 Milestone Deliverables

Your team must demonstrate a functioning integrated final system (140 points) that consists of:

- Working system prototype from Milestone 2
- Additional code implementing the required features from the Failure Alert Subsystem (20 points)
- Additional code implementing integration and general features from the five subsystems (110 points). A points cap applies to each subsystem as follows for general and integration features implemented (caps apply based on where the feature is specified):
  - Subsystem 1 (Approach Height Detection Subsystem): 30 points
  - Subsystem 2 (Tunnel Ave Control Subsystem): 10 points
  - Subsystem 3 (Over-height Exit Subsystem): 25 points
  - Subsystem 4 (Tunnel Height Detection Subsystem): 30 points
  - Subsystem 5 (Failure Alert Subsystem): 15 points
- Circuit Diagrams for all implemented features of the project. You will not be allowed to demonstrate circuits for which there are no circuit diagrams submitted. Please note that omitted circuits will result in deductions of relevant points earned from Milestone 2.
- Full system integration (of all subsystems) as a single code-base (meaning a single code launch execution point) and operating on a single Arduino (10 points).

Points caps are applied to each subsystem's general and integration features - you can implement more than the cap amount of points but you will at most earn that cap amount from the subsystem.

Deductions will only be applied after the total points earned is calculated.

Note that for full system integration marks, your code must execute from one single starting point (i.e. running the main.py file will start the entire system) and run off a single Arduino for your project demonstration.

As part of your submission, your team must demonstrate the built system's functionality to your demonstrator in your Week 12 class.

5.3.2 Milestone Weighting

This milestone is marked out of 140 points.

Marking Rubric for features:

- Functional and working without errors: 100% of points awarded
- Contains errors but can demonstrate was working in submitted video: 66% of points awarded
- Contains errors but can explain how to fix: 33% of points awarded
- Required or selected feature not attempted: 33% of points deducted
- Attempted but does not work, cannot explain: 0% of points awarded

Please note marks for the milestone will not be finalised until completion of the full project, after application of the Viva and PAF scaling (see section 4.2.6.3 and 4.3 for more details on scaling).

5.3.3 Submission Format

The circuit diagrams must be in PDF format (.pdf), in a single file containing one circuit per page, and can be landscape A4 or portrait A4. You should have one circuit for each logical component. It is strongly recommended that you complete the circuit diagrams early so that your manager can review and give you feedback on them.

The code submitted must be in python format (.py), and can be split across multiple files if needed. Note that for integration marks, your code must execute from one single starting point (i.e. running the main.py file will start the entire system). In addition, you must have attempted all features.

Teams may choose to include a short video in .mp4 format (under 500MB) demonstrating the prototype functionality being assessed in their moodle submission. This will serve as a backup for assessment if the demonstration during class fails. If the file is too big, upload the video to your google drive and **include the link to the specific videos** in your submission as a text file.

Put your project code (.py files) and circuit diagram (single .pdf file) in a folder named TeamNNN, replacing NNN with the code for your team. Then ZIP the folder up and name it TeamNNN-M3.zip, replacing NNN with the code for your team, i.e. if your team is MS02, your filename should be TeamMS02-M3.zip (max zip file size 500MB).

5.3.4 Testing

During your scheduled Practical Class in Week 12, your team will be pre-assigned a timeslot (during week 11's prac class) to demonstrate your full functional prototype, please see section 4.2.6.2 for more details. Each group member will also be required to sit an individual interview to confirm understanding of the submitted prototype, please see section 4.2.6.2 for more details and the rubric.

5.3.5 Deductions

The client defines an “inappropriate current” as a current that is either too low for consistent operation or too high for the components used (risking damage). A breakdown of all key deductions can be seen in the table below.. Please discuss any implementation requirements with your manager if you have any questions.

Category	Description	Deduction
Overall	No circuit diagram submitted for a feature.	Affected feature will not be assessed (no attempt given)
Overall	Other hardware components besides the allowed hardware components are used (items not provided in your project kit).	The project will be marked once the offending hardware components are removed - they are not to be replaced.
Overall	Non permitted packages, methods and techniques used.	Offending code to be commented out.
Overall	Code or circuit missing for a feature marked in Milestone 2	100% of the points earned for that feature in M2 is deducted.
Circuits	Circuit Diagram does not match physical circuit submitted	Late submission penalty of two days (10%) or cap of feature mark to 33%
Probe Points and Appropriate Current	● Probe point (terminal header + jumper header pins) is not included for measuring voltage on the failure alert subsystem timer ICs trigger/output pins. ● Inappropriate current for any LEDs or buzzers	-5
General Code Quality	● Console print statements are unclear, have insufficient information, or are poorly formatted. ● Console alerts are at an inappropriate rate (i.e. spams the console / prints multiple times per second). If code is used to clear the console, this deduction will automatically apply. ● Code crashes, does not exit cleanly, has an unhandled exception, uses unspecific exception handling, or exits with an error message.	-5
Physical Circuit Quality	● Physical circuit doesn’t follow wire colour conventions, ● Physical circuit doesn’t use power/ground rails appropriately ● Physical circuit uses spaghetti wiring (except for wires to the Arduino).	-5
Circuit Diagram Quality	● Circuit Diagram doesn’t meet required standards ● Circuit Diagram components are incorrectly oriented. ● Circuit Diagram is correctly labelled with unique identifiers and values. ● Circuit Diagram uses wrong component shapes (must use shapes from the library provided).	-5
Coding Quality	● Code doesn’t meet coding standards ● TODOs left in code ● Variables not in lowerCamelCase ● Functions not in snake_case ● Indentation not 4sp = 1 tab ● Function header documentation in wrong location, format, incomplete, missing or not present ● File header documentation in wrong format, incomplete, missing or not present	-5

5.3.6 Demonstration

Each group will be allocated a time slot within the 3 hour practical session in which they will need to demonstrate their prototype.

You will have 35 minutes for demonstration, so your group will need to arrive a minimum of ten minutes before your presentation time to prepare and finalise any set up. You should come to class with **your project fully assembled** as no time is given to build your circuits. Groups which are late will lose the time available to demonstrate their prototype and any items which cannot be demonstrated in the given time will not be assessed.

5.3.7 Viva (Interview)

To confirm each team member's understanding of the built prototype each member of the team will be individually interviewed about what has been developed. You will be required to answer questions based on both the software and hardware components the group has developed. Thus it is important to make sure you understand all parts of the prototype, even sections you did not individually work on.

Total project marks will be scaled based on the level of understanding demonstrated in the Viva, with the full 10 marks being no change and a reduction as the category code drops (see the table below for the marking criteria).

If you miss (do not attend) your project interview, you may receive a 0.

Category Code	Category Description	Detailed Description
10	Complete Understanding	The student has **clearly prepared** and **understands the code/circuits**. They can answer the questions correctly and concisely with little to no prompting.
9	Tolerable understanding	The student is **reasonably well prepared** and can **consistently provide answers** that are **mostly correct**, possibly with some prompting. The student may lack confidence or speed in answering.
8	Selective understanding	The student gives answers that are **partially correct or can answer questions about one area correctly** but another not at all. The student has not prepared sufficiently.
7	Trivial understanding	The student may have seen the code/circuits before, and can answer **something partially relevant or correct** to a question but they **clearly can't engage in a serious discussion of the work**.
0	No Understanding	The student has **not prepared**, **cannot answer even the most basic questions** and likely has not even seen the code/circuits before, or did not attend the interview.
AB	Absent	The student has **not attended the interview** for the project.

5.3.8 ITP Metrics Peer Assessment

In Weeks 4, 7 and 12, students will be required to login to ITPMetrics.com and complete a peer-assessment of their teammates, to give feedback to each other on their teamwork performance. This comes in the form of rating your peers on some aspects of teamwork, and providing some comments. Once this is done, the system will compile and generate some written feedback to students, which you will discuss as a team in Weeks 5 and 9, and use to improve your behaviour and performance going forward. You will explore these aspects of teamwork much more deeply in ENG1012.

The first two peer assessment factors (PAF1/PAF2) are not used to scale your performance in the project. **After the final peer assessment in Week 12**, the **system will generate a "Peer Assessment Factor" (PAF3),** which will be used to **help moderate students' individual marks**. **The PAF will be used to scale the total marks of the project to indicate overall contribution.** The PAF can range from 0 to 1.1 and is based on your teammate's ratings of your performance relative to the average rating of the entire team. Essentially, if everyone is happy, everyone receives a PAF of 1. If a team member receives a consistently higher or lower rating than the average of the team, they would receive a PAF of higher or lower than 1.

Students that do not complete their ITP Metrics peer assessments will receive a maximum PAF of 0.9.

6 System Features

This section outlines the functional requirements of the system, divided into five subsystems. The system should continue to run, uninterrupted, until the user chooses to exit (i.e. via KeyboardInterrupt). For your convenience, the features in each subsystem are color coded as follows:

- Required features are shown in **blue highlights**.
- General features are shown in normal text (no highlight).
- Integration features are shown in **yellow highlights**.

Milestone 3 Feature selection tips

This project is designed to be accessible to students of all levels - from complete beginners to seasoned programmers and electronic hobbyists. For Milestone 3, it is straightforward to achieve a credit or even distinction level grade by simply completing the lab exercises each week and modifying them to suit the project. To achieve a high distinction, you will need to complete the system integration features that **are designed to be very challenging (i.e. for coding/electronics enthusiasts)** and thus the time involved in implementing them will be much higher (**~20-30 hours for beginners**).

Teams should very carefully consider (depending on what they're aiming for as a mark) what to implement. It is possible to get up to a high Credit or low Distinction for Milestone 3 **without attempting any integration features at all**. You can read more about partial or complete integration requirements in the relevant section below. If in doubt, make sure you discuss your team's implementation plan ahead of time with your team manager.

Students that do not attempt systems integration will save a lot of time (which is a very limited resource at university), and might instead consider attaining the bonus marks for completion of the Practical Learning Competencies (2% each) within the first three weeks of the assessment offering, which will apply individually to your final project mark (capped at 26%). These marks can make up for not having attempted any integration features.

6.1 Approach Height Detection Subsystem

This is the primary overheight detection system, consisting of an overheight detection ultrasonic sensor (US1), two sets of red/yellow/green traffic lights (TL1, TL2) spaced 500m apart along the road, a speaker system (PA1) and a set of two yellow warning lights (WL1). The subsystem is the yellow highlighted section of the traffic control system prior to the over-height exit on the system overview.

The Milestone 3 cap on general and integration feature points in this subsystem is 30 points.

Code	Description	Value
1.R1	Upon detection of an overheight vehicle by US1, an alert is printed to the console. The alert should include the detected height of the vehicle, as well as the date/time of detection.	5
1.R2	Upon detection of an overheight vehicle by US1, the following sequence is triggered: 1. TL1 switches to yellow for 1s, then to red for 30s. 2. At the same time as (1), after a 1s delay, TL2 switches to yellow for 1s then red for 30s. 3. After the red lights are shown for 30s, both TL1 and TL2 switch back to green.	17
1.G1	Upon detection of an overheight vehicle by US1, WL1 turns on and flashes each yellow LED in turn at 2-5 Hz so long as TL1 and TL2 are not green.	8
1.G2	Upon detection of an overheight vehicle by US1, PA1 turns on and begins sounding a unique buzzer tone so long as TL1 and TL2 are not green.	8
1.G3	Upon detection of an overheight vehicle by US1, TL1 and TL2 stays red if US1 continues to detect an overheight vehicle. After the initial 30s of red light duration, the buzzer tone on PA1 changes to a higher frequency tone. (tone must be at least 2000 Hz higher)	12
1.G4	The data feed from ultrasonic sensor US1 is filtered for noise appropriately. (This means that random signals will be ignored and smoothed over)	3
1.I1	**This feature requires integration with subsystem three.** *This feature overrides the default behaviour above.* When TL1 and TL2 turn red as controlled by US1, they will stay red until US3 detects an overheight vehicle passing (hence exiting the system), and both US1 and US3 no longer detect an overheight vehicle.	12

6.2 Tunnel Ave Control Subsystem

This is the subsystem in control of the red/yellow/green traffic lights (TL4) as well as the pedestrian crossing with its associated red/green pedestrian lights (PL1) and pair of pedestrian crossing push buttons (PB1) on the Tunnel Ave arterial road. This subsystem is highlighted in pink on the system overview.

The Milestone 3 cap on general feature points in this subsystem is 10 points.

Category	Description	Value
2.R1	If the pedestrian push button is pressed, the following sequence should be triggered after a two second wait. 1. TL4 turns yellow for 2 seconds, then red for 5 seconds. 2. While TL4 is red, switch PL1 to green for 3s, then flashing red for 2s before resetting to solid red. 3. TL4 turns back to green.	18
2.R2	If the pedestrian push button PB1 is pressed, show this information on the console once (until the sequence above is run). (i.e. you should ensure that holding down or spamming the push button does not trigger multiple console prints).	5
2.G1	After the pedestrian push button PB1 is pressed (and 2.R1 executes), wait at least 30 seconds before allowing 2.R1 to execute again.	7
2.G2	The pedestrian push button PB1 is debounced using hardware components such that a clear signal is sent.	3

6.3 Over-height Exit Subsystem

This is the subsystem controlling the set of the red/yellow/green traffic lights (TL5), a light dependent resistor for detecting day/night (DS1), a set of two floodlights (white LEDs) (FL1, FL2) as well as the vehicle presence detection ultrasonic sensor (US3) on the over-height exit off the A102. This subsystem is highlighted in purple on the system overview.

The Milestone 3 cap on general and integration feature points in this subsystem is 25 points.

Category	Description	Value
3.R1	Upon detection of an overheight vehicle by US3, the following sequence should trigger: 1. TL5 should go from red to yellow for 2s 2. TL5 then turns green for 5s 3. TL5 then turns back to red	15
3.G1	If the ultrasonic sensor US3 continues to detect an overheight vehicle after TL5 has stayed green for 5s, the following sequence runs instead: 1. TL5 flashes green at 2-5 Hz continuously until US3 no longer detects an overheight vehicle 2. TL5 turns back to red (otherwise (1) will continue).	12
3.G2	The data feed from ultrasonic sensor US3 is filtered for noise appropriately. (This means that random signals will be ignored and smoothed over)	3
3.G3	If the ultrasonic sensor US3 detects an overheight vehicle and the LDR DS1 detects that it is night time, the following events occur: 1. Turn FL1 and FL2 on and this stays on while US3 detects an overheight vehicle 2. Turn FL1 and FL2 off	5
3.I1	**This feature requires integration with subsystem two.** *This feature overrides the default behaviour.* When TL5 turns yellow, execute the following sequences: ● TL4 should turn yellow for 2s, then red for 5s, then back to green. ● If PB1 is pressed, PL1 turns green immediately while TL4 is yellow or red.	12
3.I2	**This feature requires integration with subsystem one.** *This feature overrides the default behaviour.* Upon detection of an overheight vehicle by US3, execute the following sequences: ● If US3 no longer detects an overheight vehicle, set TL1 and TL2 back to green, and turn off WL1 and PA1.	8

6.4 Tunnel Height Detection Subsystem

This is the “last chance” overheight detection system, consisting of an overheight detection ultrasonic sensor (US2), a set of red/green traffic control lights (TL3), and two sets of two red warning lights (WL2) which are used to indicate tunnel closure when flashing. This subsystem is highlighted in green on the system overview.

The Milestone 3 cap on general and integration feature points in this subsystem is 30 points.

Category	Description	Value
4.R1	Upon detection of an overheight vehicle by US2, the following sequence should trigger: 1. TL3 should turn from green to red immediately. The system can only reset to ‘normal state’ if an overheight vehicle is no longer detected by US2. (i.e. TL3 turns back to green)	10
4.G1	Upon detection of an overheight vehicle by US2, the following sequence should trigger: 1. WL2 turns on and flashes each red LED in turn at 2-5 Hz so long as US2 detects an overheight vehicle.	8
4.G2	The data feed from ultrasonic sensor US2 is filtered for noise appropriately. (This means that random signals will be ignored and smoothed over)	3
4.I1	**This feature requires integration with subsystem one.** *This feature overrides the default behaviour.* Upon detection of an overheight vehicle by US2, execute the following sequences until US2 no longer detects an overheight vehicle: ● TL1 and TL2 turns red ● PA1 starts sounding a unique tone ● WL1 turns on and flashes each yellow LED in turn at 2-5 Hz	12
4.I2	**This feature requires integration with subsystem two.** *This feature overrides the default behaviour.* Upon detection of an overheight vehicle by US2, execute the following sequences until US2 no longer detects an overheight vehicle: ● TL4 turns red	5
4.I3	**This feature requires integration with subsystem one and subsystem three.** *This feature overrides the default behaviour.* Upon detection of an overheight vehicle by US2, execute the following alternate behaviour: ● The system reset to ‘normal state’ now relies on: ○ US3 detecting an overheight vehicle exiting ○ US1 and US2 no longer detecting an overheight vehicle ● If the condition above is met, allow TL1, TL2, TL3, TL4 to go back to green, and allow TL5 to go back to red.	12

6.5 Failure Alert Subsystem

This is the subsystem that operates in case of power failure in order to shut down the tunnel for safety reasons. It consists primarily of a backup battery pack, an override switch (OS1), a speaker system (PA2), and has overarching control of all the red traffic lights on the A102 (TL1, TL2, TL3, WL2). This subsystem is highlighted in orange on the system overview.

The Milestone 3 cap on general and integration feature points in this subsystem is 15 points.

Category	Description	Value
5.R1	This subsystem is powered by an external battery supply and must not be powered by the Arduino's pins. Upon detecting a loss of power to the Arduino (by monitoring the Arduino's power pins), PA2 should start sounding an alert tone between 1000 Hz and 3000 Hz immediately. A 555 timer must be used to generate the tone. The alert tone should be clearly audible (not soft) but not excessive (not too loud), and calculations must be supplied to support selected volume.	10
5.R2	The subsystem maintains the alert tone for 3s after power is restored, after a power outage has occurred. (This means the tone should continue to sound for 3s after power is restored after a power outage, and not impact the immediate start of the alert tone upon detection of a power outage)	10
5.G1	If the override switch OS1 has been turned on, enable the Failure Alert subsystem (act as if a power outage has been detected)	8
5.G2	The override switch OS1 is debounced using hardware components such that a clear signal is sent.	3
5.I1	**This feature requires integration with subsystem one.** *This feature overrides the default behaviour.* If the Failure Alert subsystem is triggered (on a power outage), execute the following actions: ● TL1 and TL2 turns red until power is restored	8
5.I2	**This feature requires integration with subsystem four.** *This feature overrides the default behaviour.* If the Failure Alert subsystem is triggered (on a power outage), execute the following actions: ● TL3 turns red until power is restored ● WL2 turns on and flashes each red LED in turn at 2-5 Hz using a 555 timer to generate the flashing signal until power is restored	8

6.6 Overall Integration

Warning: Complete Integration is hard and can be extremely time consuming!

It is recommended that teams only work on integration if they're confident - without complete integration, students must be able to demonstrate each subsystem's features as described.

There are **two** levels of integration for this project.

Partial Integration points are provided to you in each subsystem, you are provided with integration features - you can integrate the listed subsystems and implement those features to obtain those integration marks. Please note that you will need to run the integrated subsystems off a single Arduino, both in software and in hardware.

Complete system integration requires all of the subsystems to interact with each other, regardless of which integration features your team decides to implement. This means that your team may only use a single Arduino and codebase for the entire project.

Using one Arduino will require you to make careful decisions around the choice and usage of pins. Alternatively, as demonstrated in practice classes, you may consider using multiplexers such as the shift register (outputs) or an R-2R ladder (inputs) arrangement to manage pin usage.

You should note that it is often the case that you may 'break' existing functionality while working to integrate multiple subsystems. This usually happens if your team attempts to rush the process, which may negatively impact your marks. Therefore **in the final days before submission, it is better not to attempt** integration, as **trying to do so at the "last minute" will often result in accidental breakages to your system**.

Regardless of integration level, the base expectation is that your team is able to demonstrate each subsystem on its own (assuming no integration was done). If your team has implemented *some* integration features, then those subsystems must work together (run from the same code launch, same arduino) as described in the feature.

Chat with your team manager if you are unsure of how to approach this.