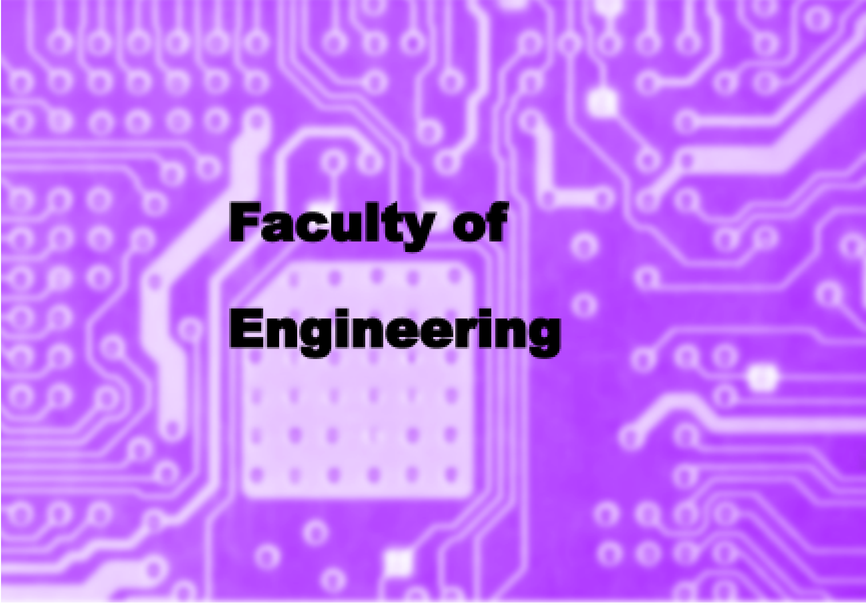
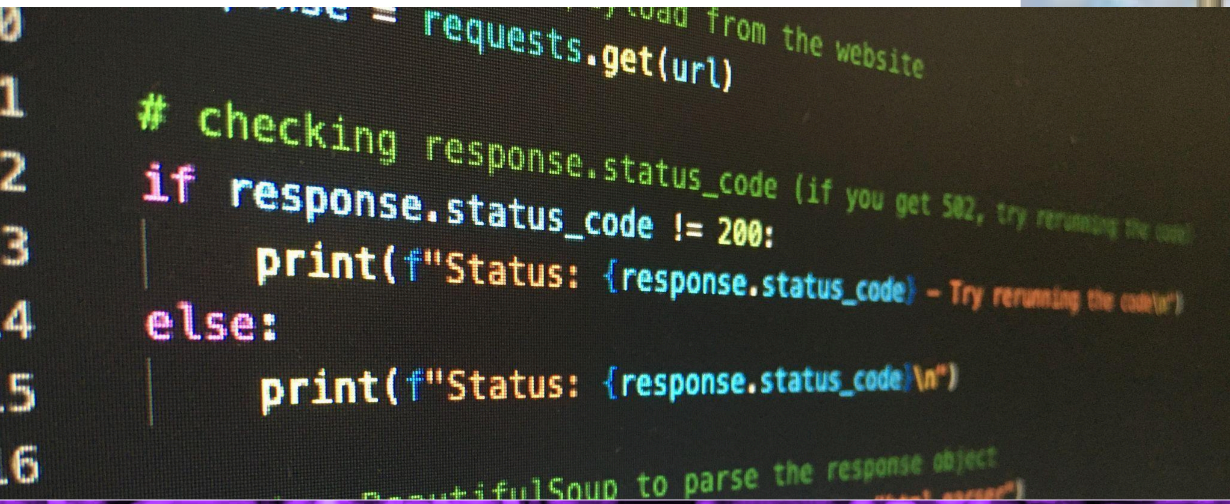


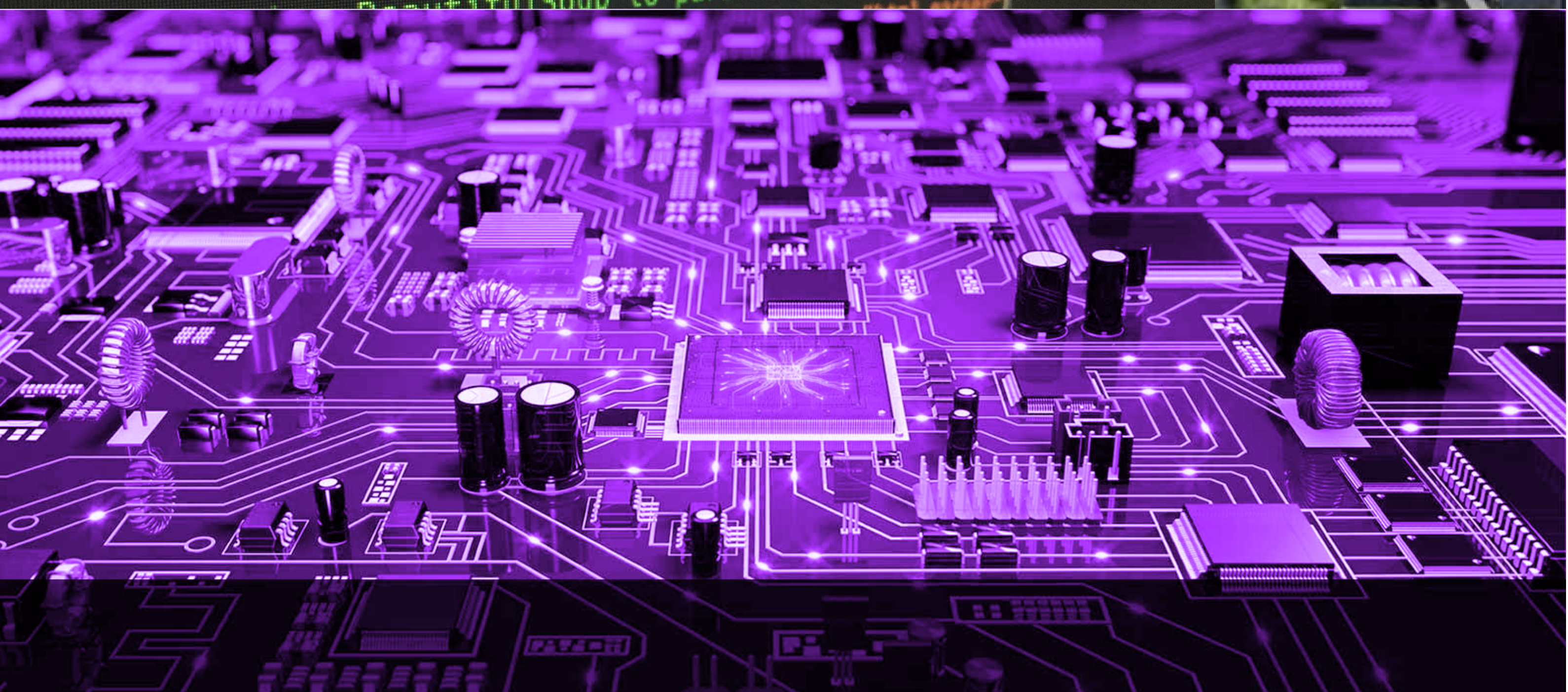
Faculty of Engineering

ENG1013: Engineering smart systems

Traffic Control System Project

Project Specifications





1 Introduction

“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.

Common 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. Traffic control systems are also used in other contexts such as at airports, in railway signalling, on waterways, or at race courses.

2 Project Requirements Overview

Your team has been asked to design, build, and demonstrate a traffic light control system in this project. Your system will consist of subsystems. The client has broken down each subsystem into a group of required features. These make up the "minimal viable product" - this is an early version of a design that helps to validate the idea. Your team must implement these features. Most subsystems also have a group of general features, which your team may choose from for the final submission.

The project will be graded out of 200 points, with your final individual score scaled to 20% of your total mark for ENG1013. Results for the project do not get finalised until all the milestones are complete.

2.1 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), plus single core wire and additional breadboards. However, you may use only **ONE** Arduino to attain maximum points for an integrated system. You may not use any other hardware components.

2.2 Software Requirements

You may develop your software in any 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 are not permitted, and their use will result in a points deduction. Students are not allowed to use Classes and assert statements.

In order to consider your system integrated, all code must run from a single base file - you may have multiple self-written modules (other files your team created) that are called from this base file.

3 System Specifications and Overview

Your task is to design a traffic control system for a 1-way road that has a pedestrian crossing and a minor road joining onto the main road using a controlled traffic light junction. The vehicle (road) traffic, the pedestrian traffic, and the minor road traffic must all be controlled via sets of traffic lights. The pedestrian lights have red and green lights, and the vehicular traffic lights have red, amber (yellow), and green lights. The specifications below describe the required system behaviour.

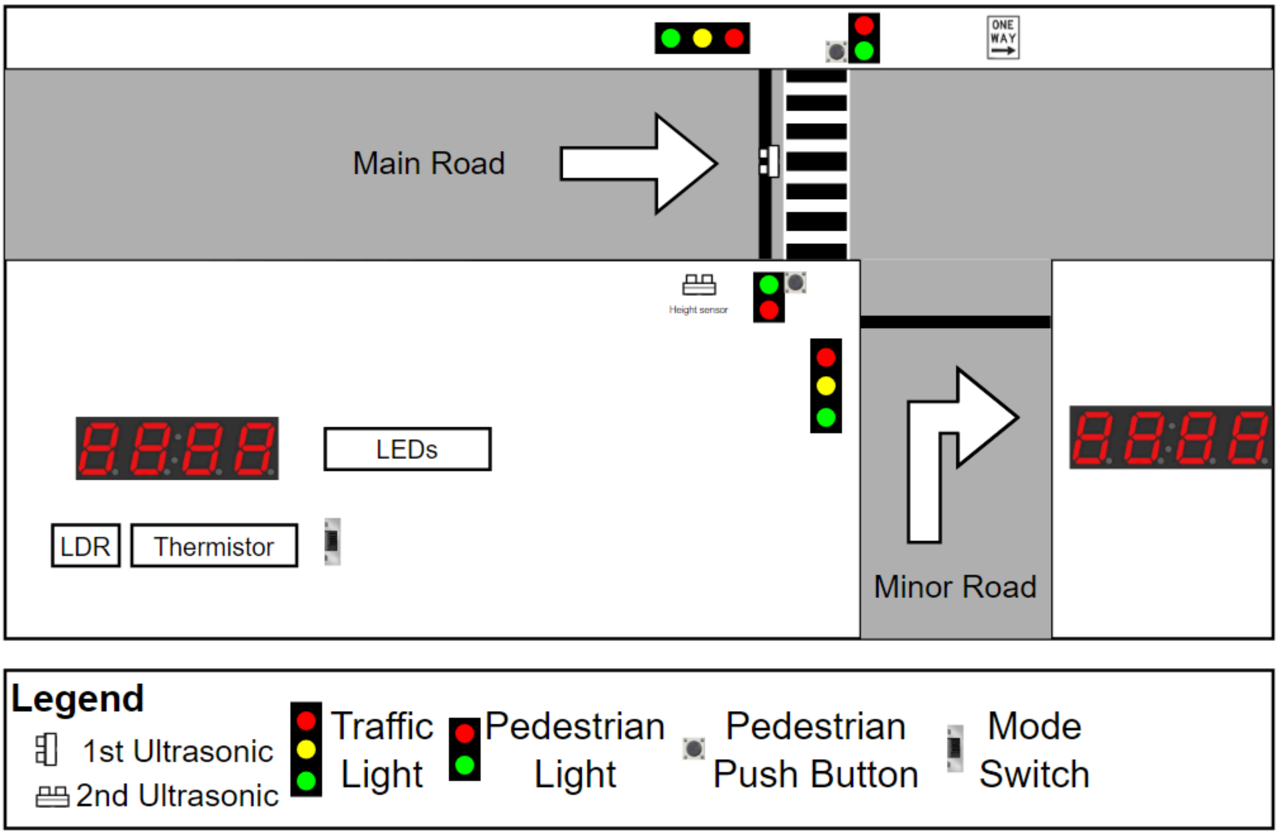


Figure I: Model Road, showing all parts (MVP + FP).

The system has been broken down into two parts, the Minimum Viable Product (MVP) and the Final Product (FP).

The MVP will detect the presence of vehicles using an ultrasonic sensor, and will implement the two sets of traffic lights for both the main and minor road junctions, along with the pedestrian lights for the crossing. The pedestrian crossing has a set of push buttons to detect the presence of any pedestrian.

The FP contains additional optional features such as a light-dependent resistor (LDR) for a day-night detection, vehicular height using a second ultrasonic sensor, and a thermistor that acts as a weather (temperature) sensor.

In this document, the lights to control the vehicular traffic will be referred to as vehicle traffic lights, and the lights to control the pedestrian traffic will be referred to as pedestrian traffic lights.

3.1 System Overview

We have divided the system's functionality into five key subsystems. Information is shared between the subsystems to form a cohesive system. These are:

1. Control Subsystem
2. Services Subsystem
3. Inputs Subsystem
4. Outputs Subsystem
5. Failure Alert Subsystem

The system should have three operating (3) modes, which should be controlled by the **services subsystem**.

- **"Normal Operation"** - traffic system is in normal operation
- **"Maintenance Adjustment Mode"** - traffic system is suspended in maintenance mode
- **"Data Observation Mode"** - traffic system is suspended and data is graphed/visible to user

The **control subsystem** is responsible for regular traffic monitoring to ensure the least amount of car and pedestrian delay. It interacts with the **inputs subsystem** which has a series of sensors that provide data to the system, as well as the **outputs subsystem**, which displays alerts, alarms, information and sends signals to an external system that would change the traffic lights based on the presence of traffic. Your team should select and program a set sequence of lights. More information on each of the subsystems can be found below.

The "control" subsystem controls the fundamental system operation mode, and provides the main polling (operating) loop of the system. It contains the functions of the "inputs", "outputs", and "failure alert" subsystems to manage all physical (real world) signals interacting within the system.

The "input" subsystem measures the presence of cars in the system at an appropriate polling rate. The "output" subsystem processes this reading and sends signals to turn on the required lights to ensure the traffic is processed appropriately. It also issues any alerts or alarms necessary. The input and output subsystems can also be added upon for an additional road or cause changes based on time of day (level of light).

The "Services" subsystem links the entire system together. It allows the user to switch between the three required modes. In addition, this subsystem directly provides for the system settings to be modified by entering the correct personal identification number (PIN). If the PIN is entered incorrectly multiple times in a row, the user will be locked out of the system settings. The "data observation mode" is also accessed from the services subsystem. It processes the data stored by the control subsystem to provide user-relevant human outputs. These can be supplied on a seven-segment display, console "print" statements, and as graphs of key variables.

Finally, the "failure alert" subsystem enables the traffic light system to safely alert users to a power failure. The alarm is powered independently to the rest of the system, and upon system power failure, a variable frequency buzzer is activated.

The minimum viable product **requires** the integration of base features in control services and inputs subsystems. You should discuss the necessary functionality and information flows between each subsystem with your manager. The final product should contain the fully functional and integrated minimum viable product, along with additional features as selected from the options available.

4 Deliverables

Before authorising the final product, your manager needs to see several intermediate design stages: the initial design and a prototype of the minimum viable product. You will complete these in the form of several milestones during the semester, before demonstrating the final prototype and answering questions related to its design and operation in week 12. Please be mindful that the milestones do not directly correspond to the subsystems; a range of functionality is required for each milestone.

There are four project milestones:

1. Delivery of a system interaction block diagram and pseudocode for the minimum viable product (30 points)
2. Minimum Viable Product Demonstration (40 points)
    a. Checkpoint to test the basic software functionality of the software minimum viable product. (up to -10 points taken from the mvp score)
    b. Delivery of the minimum viable product (software and hardware) (40 points)
3. Full system prototype demonstration and viva (oral interview) (100 points)
4. Video presentation and comments on project extensions: reflection, machine learning & data ethics (30 points)

For more detailed information on the submission of each milestone, please refer to the associated milestone sections. Please see below for detailed information on each milestone's required and general features.

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. Should any member of the team not be present during the demonstration, they will need to apply for special consideration to receive the points. Depending on the situation, an additional interview may be conducted. Deductions apply for not meeting coding or circuit standard requirements.

4.1 Minimum Viable Product (MVP)

The MVP describes a simplified prototype that contains four subsystems, control; services, inputs and outputs. You will develop an integrated control and services system that reads from limited input sensors, and produces the necessary control displays and outputs as a proof of concept before implementing the full set of features.

4.1.1 Control Subsystem

The control subsystem for the MVP includes function calls to the hardware features from the system input and output subsystems.

Category	Description	M1 Points	M2 Points
System	The normal operating mode consists of a polling loop.	2	2
System	The time to complete one polling loop cycle is measured and printed to the console.	2	2
System	The time between polling cycles (time between each cycle of 'sensor polling') is between 1 and 5 seconds.	2	1

4.1.2 Services Subsystem

The services subsystem provides the main system structure and menu - This is what the system boots into on launch. It is up to your team how you wish to re-enter the menu in the services subsystem if you are operating a different functionality (i.e. simply restarting code to re-enter or return to the menu is not appropriate). Your system should be able to run on any computer and be operating system independent.

You can select any information to present to the user in the services subsystem, however - you must be able to justify your choices. The minimum requirement is to use the graph to display the last 20 seconds of traffic distance measurements.

Category	Description	M1 Points	M2 Points
System	The system menu allows users to select between Normal Operation, Data Observation Mode, and Maintenance Adjustment Mode.	2	1
Settings	The Maintenance Adjustment Mode requires the correct pin from the user before viewing or updating the system parameters.	2	1
Settings	New user-modified parameters are displayed, stored and used for future operations.	2	2
Settings	The user is denied access to modifiable parameters upon entering an incorrect PIN. The user is informed that the PIN is incorrect.	2	1
Graphing	The system generates a graph of the ‘traffic distance’ for the last 20 seconds (not live), using the most recent data from the ultrasonic sensor in the integrated polling loop. The graph is well formatted with a legend, both axes are labelled, and have appropriate titles.	2	3
Seven Seg	The seven-segment display is used to: ● display multiple four-digit alphanumeric messages. ○ the message must be relevant to the project ○ the messages must not be hard-coded ■ your function should be able to accept any message and display it on the seven-segment display ○ the message must be stable and clearly visible	2	7
Graphing	Your code should have an appropriate response when insufficient data exists at the point when the graph is generated.	2	2

4.1.3 Inputs Subsystem

The inputs subsystem is primarily responsible for the interface with reading sensor data. The ultrasonic is required to read the distance to the oncoming vehicle.

Category	Description	M1 Points	M2 Points
Ultrasonic	The ultrasonic sensor records a distance measurement at an appropriate polling rate of <5s. Your team should be able to justify the choice of polling rate. The distance measurements are filtered so that the output is reasonable and stable.	2	1
Ultrasonic	The rate of distance change (dV/dt) that should trigger system alerts for your project is experimentally determined and justified with results.	Not assessed	2
Push Button	The pedestrian push button triggers a single clean signal (de-bounced in software) when pressed, and indicates the presence of pedestrians.	2	2

4.1.4 Outputs Subsystem

The outputs subsystem controls the state of the traffic lights. Information about the current stage of traffic operations is also displayed on the console.

Category	Description	M1 Points	M2 Points
Traffic Operation	The sequence of traffic light operations is as follows: 1. Stage One: 30 seconds ○ Main Road Traffic Lights: Green ○ Side Road Traffic Lights: Red ○ Pedestrian Lights: Red 2. Stage Two: 3 seconds ○ Main Road Traffic Lights: Yellow ○ Side Road Traffic Lights: Red ○ Pedestrian Lights: Red 3. Stage Three: 3 seconds ○ Main Road Traffic Lights: Red ○ Side Road Traffic Lights: Red ○ Pedestrian Lights: Red 4. Stage Four: 30 seconds ○ Main Road Traffic Lights: Red ○ Side Road Traffic Lights: Green ○ Pedestrian Lights: Green 5. Stage Five: 3 seconds ○ Main Road Traffic Lights: Red ○ Side Road Traffic Lights: Yellow ○ Pedestrian Lights: Flashing Green at 2-3 Hz 6. Stage Six: 3 seconds ○ Main Road Traffic Lights: Red ○ Side Road Traffic Lights: Red ○ Pedestrian Lights: Red 7. Repeat from Stage One	2	5
Traffic Operation / Console	The current stage of traffic operation is shown on the console once per stage in the current sequence.	2	1
Distance/Console	The system displays the distance to the nearest vehicle in two decimal cm readings on the console, at most once every 1-3 seconds.	1	1
Pedestrian Presence	The system displays the presence of pedestrians on the console as the total number of presses at the beginning of stage three. This value resets every time the traffic operation sequence restarts at stage one.	1	1

4.1.5 Integration

System integration requires the subsystems to interact with each other, such that data received on the input causes change on the output and with this system being editable via the service subsystem. Therefore, your team may only use a single Arduino and run off a single base file (though other files can be imported as modules) for your integrated system.

Due to the large number of Arduino pins required to run a seven segment display however, for the MVP it is **not** required to be integrated and can run from a separate file. This **will** be required for the final prototype so no marks will be deducted for its integration **unless** functionality is compromised.

Category	Description	M1 Points	M2 Points
Integration	All hardware except the seven segment display must run off one Arduino. All software must execute off one base file (i.e. only hit 'Run' once)	Not assessed	5

4.1.6 Milestone 1

This milestone is due at the end of Week 4.

Milestone Deliverables:

1. System interaction diagram of the MVP. The diagram should cover all of the relevant marked requirements from the MVP, showing the interaction between 'blocks' of functionality and the relationship (lines) between these and various subsystems. Refer to Moodle and your lab manual for examples.
2. Low Level (detailed) pseudocode for all software features required for the MVP.

Milestone Weighting:

In each specification the mark for the item in relation to Milestone 1 is listed as M1 points. This is out for a total of 30 points.

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).

Submission Format:

Submit your document as a single PDF file, and name it TeamNNN-M1.pdf, replacing NNN with the three character for your team number, i.e. TeamMH05-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 pseudocode pages being A4 Portrait.

Restrictions:

Your team contract must be submitted, and accepted (graded) by your demonstrator in order for your team to submit. If you submitted your contract last second instead of as instructed in class, you will need to chase your demonstrator to mark it off, otherwise you will not have access to the Milestone submissions.

4.1.6.1 Deductions

Category	Description	Deduction
Overall Submission	Incorrect submission format (filetype, structure)	-3
Pseudocode	Pseudocode contains jargon or is of bad quality. Note: Jargon includes coding syntax.	-6

4.1.7 Milestone 2

This milestone contains two parts. Part 1 is a checkpoint in Week 6's practical class, and Part 2 is a submission at the end of Week 8, with marking done in class in Week 9's practical class.

4.1.7.1 Part 1 - Checkpoint (Week 6)

In week **six** during your practical class, there is a **checkpoint** that is marked as a -10 point (max) deduction from your Milestone 2 submission. This means that if you fail all items on the checkpoint, you will lose ten points from your Milestone 2 submission.

Milestone Deliverables:

The team must demonstrate code which generates a working polling loop from the control subsystem, integrated with the three required modes from the services subsystem, with placeholder functions for any relevant inputs/outputs that produce console prints to demonstrate that it works. The input/output functional **do not** need to connect to hardware, only exist with print statements indicating the logical flow has been determined.

Specifically, we will be checking that:

- A polling loop exists.
- The polling loop works, and has placeholder functions which poll sensors as needed.
- The polling loop is integrated with the menu from the services subsystem and has the three required modes in the menu. All the modes are accessible in the menu.
- Placeholder functions exist for all relevant inputs and outputs in the MVP.
- All placeholder functions have appropriate print statements as required.

Submission Format:

You do not need to submit anything to Moodle for this checkpoint. This is assessed during class.

Testing:

During your scheduled Practical Class in Week 6, we will check your work to see if you have met the checkpoint requirements.

4.1.7.2 Part 2 - MVP Submission (Week 8)

Milestone 2 is due at the end of Week 8, with marking done during class in Week 9.

Milestone Deliverables:

The full functional MVP prototype will need to be demonstrated so that its functionality can be tested. This will include the control, service, input and output subsystems as specified in the project brief under MVP. This Milestone will include Moodle submissions and an in class presentation of the built prototype (more information on testing of the prototype is under the testing subheading).

By the Milestone 2 due date on Moodle the Circuit Diagrams for all the minimum viable hardware design requirements specified by the project brief, and Python Code written to implement the minimal viable hardware design requirements specified by the project brief must be submitted. Files should be sensibly named.

Teams may choose to include a short video demonstrating the prototype functionality being assessed in their moodle submission. This will serve as a backup for assessment if the demonstration during class fails (see deliverables/section 4 for details on how this will be marked.

You will need to select the features you will implement for Milestone 3 and submit this via the SYSTEM. This is due at the same time as your Milestone 2 submission. Failure to submit this via the SYSTEM will incur late submission penalties for Milestone 2. One student should submit the feature selection in the SYSTEM on behalf of the team.

Milestone Weighting:

In each specification the mark for the item in relation to Milestone 2 is listed as M2 points. This is out for a total of 40 points.

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).

Submission Format:

Submit the feature selection via the SYSTEM. (Links in Project under Milestone 2)

For full submission marks your code submission should meet the coding standards provided. The circuit diagrams should be titled and should be one circuit per page, with each page being A4 (or larger as necessary) in size, make sure that your circuit diagrams are readable.

Your folder may contain additional .mp4 videos demonstrating functionality as required.

Put your project code and circuit diagrams in a folder named TeamNNN, replacing NNN with the three character for your team number, i.e. TeamG05. Then ZIP the folder up and name it TeamNNN-M3.zip, replacing NNN with the three character for your team number, i.e. TeamG05-M3.zip

Testing:

During your scheduled Practical Class in Week 9, we will test the MVP 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, you will not receive marks**, so make sure you arrive on time.

4.1.7.3 Deductions

Console “print” statements are used to present all essential information to the user. These prints must be formatted in a manner that makes the meaning of each statement obvious to the user and generated at a useful and comfortable rate for the user to read. Points will be deducted if the console print statements are insufficient. Similarly, it is important to have appropriately formatted graphs to improve readability.

Additional deductions based on the good coding/circuit layout practice requirements also apply to this submission. 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). Your team and manager will use your system's probe points to measure the current of your components.You are required to use jumper pins, not jumper wire for the probe points. Discuss the implementation requirements with your manager if you have any questions.

Additional deductions based on the good coding/circuit layout practice requirements also apply to this submission. A breakdown of all key deductions can be seen in the table below.

Category	**Description**	**Deduction**
General	Console print statements are unclear, have insufficient information, or are poorly formatted.	-1
	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.	-1
	The code cannot return to the main menu from anywhere in the polling loop during normal operations.	-1
	Code crashes, does not exit cleanly, has an unhandled exception, uses unspecific exception handling, or exits with an error message.	-1
	The seven segment display current or voltage is too high for the display segments.	-1
	Probe points (terminal header + jumper header pins) are not included for measuring current and voltage on the seven-segment display.	-1
Circuit	Circuit Diagram components are in the wrong orientation, power source and ground not oriented correctly (source at top left, ground bottom right), labels not oriented horizontally and readable.	-3
	Circuit Diagram doesn't use clear, well labelled nodes (including missing labels), or are labelled incorrectly, or is missing the title block, or doesn't use the correct component diagram	-3.5
	Physical circuit doesn't follow wire colour conventions, doesn't use rails appropriately, or uses spaghetti wiring.	-2
Coding	TODOs left in code	-2
	Variables not in lowerCamelCase	-2
	Functions not in snake_case	-2
	Indentation not 4sp = 1 tab	-2
	Function header documentation in wrong format, incomplete, missing or not present	-2
	File header documentation in wrong format, incomplete, missing or not present	-2
	Non permitted imports exist, or non permitted techniques used.	-10 ea

4.2 Final Product

As a team, you can choose to implement any features in your final design, however, you should consider both the restrictions in place for the number of points in each subsystem and the interdependence of certain features. For example, the services subsystem has a maximum contribution of 10 points, and you cannot obtain points for a temperature response, if you have not chosen to implement a thermistor. If you are unsure, ask your manager.

Your team **must complete** the Failure Alert subsystem's minimum functionality (subsystem 5) -- highlighted **in blue**. You cannot "top up" the points allocated to these features with other features. To obtain points from features selected in Milestone 3 that extend on previously implemented functionality, the original functionality from the MVP must work during your Milestone 3 test.

Some sections may have multiple options with some implementations being worth more points than others or may have a set number of points available for the multiple steps required for the implementation. Features that require a choice of implementation are highlighted in a darker colour and only one option may be selected. The number of points for each section if specified will be specified in ().

4.2.1 Services Subsystem

Your team may enhance the functionality of the system menu to improve the security of the smart system. Likewise, your team may significantly enhance the functionality of the seven-segment display to improve the data sharing capabilities of the traffic light system. You may also wish to implement additional graphs or save the graphs for long-term analysis. Note, if scrolling messages are selected - all messages regardless of length must scroll.

A maximum of 15 points from this subsystem will contribute to your M3 points total. You can take on more than the maximum points in value from this subsystem but only the maximum will apply to the total.

Category	Description	M3 Points
Settings	If a user enters the PIN incorrectly three times, further access attempts (PIN entries) are denied for two minutes.	2
Settings	Admin access (ability to edit settings) times out after an appropriate period of time.	3
Settings	User-modified parameters are checked against an allowed range and only saved if valid; the user is informed if parameters fall outside this range.	3
Seven Seg	The seven-segment display shows scrolling messages.	5
Seven Seg	The seven-segment display has minimal flickering of the digits. The console displays the nominal seven-segment display refresh rate in Hz once every normal operation stage (i.e. at the beginning of each stage).	5
Graphing	Graphs are saved to a file with a logical file name upon exit.	2
Graphing	The user can view graphs of **two** other types of key information. Graphs are well formatted with a legend, both axes are labelled, and have appropriate titles.	2

4.2.2 Inputs Subsystem

Your team may improve the accuracy of the traffic system's operation by increasing the polling rate and implementing custom calibration modelling. Additional monitoring of the temperature and the presence of light may further improve the system's safety. A user can use the push button to switch states as a manual override.

A maximum of 15 points from this subsystem will contribute to your M3 points total. You can take on more than the maximum points in value from this subsystem but only the maximum will apply to the total.

Category	Description	M3 Points
Ultrasonic	A second ultrasonic sensor is used along the main road to check the maximum height of vehicles passing into this intersection. This ultrasonic sensor should be positioned at 28 cm off the surface.	3
Ultrasonic	The distance measurement function from the original ultrasonic sensor should produce a filtered value requiring a minimum of at least 1 reading every 1 second (e.g. 5 readings averaged over 4 seconds)	2
Pushbutton	The pedestrian push button is used to interrupt the traffic sequence. • If the traffic sequence is in Stage One, then this will reduce the time left in sequence to a maximum of 5 seconds. • If the traffic sequence is in other stages, no change is effected	2
Pushbutton	The pedestrian push buttons are de-bounced using hardware (DSO used to verify waveform).	2
Mode Switch	A slide switch is used to manually override the current mode - when turned on it disables normal operation mode. If the system is in normal operation mode it should exit into the menu and prevent the system from going back into normal operation mode until the switch is turned off.	2
Thermistor	A thermistor is used to read the temperature. (1) • The sensor must be calibrated to return a temperature within +/- 1C of ambient. (1) • An appropriate filter is applied. (1)	3
LDR	A light-dependent resistor (LDR) is used to measure the light levels. (1) • The sensor must be calibrated to detect the required brightness threshold effectively. (1) • An appropriate filter is applied. (1)	3

4.2.3 Outputs Subsystem

Your team may use alerts and alarms to improve the safety of the traffic system. Software and hardware options are available for both flashing LEDs and buzzers, with hardware options attracting more points than software options. Note, if multiple outputs to the seven segment display are selected, the messages should switch between each required output in sequence.

A maximum of 30 points from this subsystem will contribute to your M3 points total. You can take on more than the maximum points in value from this subsystem but only the maximum will apply to the total. **Features that require a choice of implementation are highlighted in a darker colour.**

Category	Description	M3 Points
Seven Seg	The seven segment display shows useful information about the current stage or sequence of traffic operations (more than existing MVP).	2
Seven Seg	The seven segment display shows information about the temperature (must select thermistor feature in input subsystem).	2
Seven Seg	The seven segment display shows information about the day-night cycle (must select LDR feature in input subsystem).	2
Over-height Seven Seg	A second seven segment display is used to permanently show the height of the last vehicle that passed (must select the second ultrasonic sensor in the input subsystem).	6
Over-height Buzzer	A unique buzzer tone is sounded for 2 seconds when a vehicle passes by that is higher than a predetermined height (this should be an editable system parameter). This must be implemented in hardware using a 555 timer (must select the second ultrasonic sensor in the input subsystem).	5
Over-height Flashing LED	An RGB LED is used to flash RED then YELLOW colour at an appropriate frequency (1-2Hz) continuously for 6 seconds after a vehicle passes by that is higher than a predetermined height (this should be an editable system parameter). This can be implemented • in **software** via switching the RGB LED channels on and off at the appropriate rate • in **hardware** via a 555 timer circuit to provide the frequency signals. (Must select the second ultrasonic sensor in the input subsystem)	3 Software 5 Hardware
Console	An appropriate console alert is issued where the second ultrasonic sensor detects a vehicle higher than a predetermined height (this should be an editable system parameter). (Must select the second ultrasonic sensor in the input subsystem)	1
Maintenance Mode Flashing LEDs	If the system is not in normal operations mode, the traffic lights for both main and minor roads should flash the YELLOW LED at an appropriate frequency (1-2Hz) continuously. This can be implemented: • in **software** via switching the LED on and off at the appropriate rate • in **hardware** via a 555 timer circuit to provide the frequency signal.	2 Software 4 Hardware
Ultrasonic	If the first ultrasonic sensor detects that a car is within a specific range (pick a distance, justify using experimental data, should be specified as a system parameter) when the main road traffic lights are **yellow**, extend the yellow light time for 3s.	3
Console	An appropriate console alert is issued where the first ultrasonic sensor detects a constant distance to vehicle for more than 3s while the main road traffic light is green.	1
Buzzer	Using a buzzer for the pedestrian crossing, the system should sound a unique buzzer tone in Stage Two during normal operation. You may implement the buzzer/buzzer tone: • in **software** (using a PWM pin) for 2 points; or, • in **hardware**, using a 555 timer for 4 points.	2 Software 4 Hardware

Buzzer	Using a buzzer for the pedestrian crossing, the system should sound a unique buzzer tone in Stage Five during normal operation You may implement the buzzer/buzzer tone: ● in **software** (using a PWM pin) for 2 points; or, ● in **hardware**, using a 555 timer for 4 points.	2 Software 4 Hardware
LDR	An LDR (must select LDR in input subsystem) is used to modify the traffic operation sequence in a day-night cycle. The current sequence known is the day cycle. The night cycle modifies timings for the stages as follows: ● Stage 1: 45 seconds ● Stage 4: 10 seconds This can be implemented as: ● A **software** based response (1) ● A **hardware** based response (3)	1 Software 3 Hardware
Thermistor	A thermistor (must select Thermistor in input subsystem) is used to modify the traffic operation sequence. When the temperature is above 35 degrees, increase Stage 1 and Stage 4 timings by 5 seconds each. This can be implemented as: ● A **software** based response (1) ● A **hardware** based response (3)	1 Software 3 Hardware

4.2.4 Failure Alert Subsystem

This subsystem implements an alarm system to alert users to the failure of your smart monitoring system. This alarm must consist of a variable frequency buzzer, driven by an astable 555 timing circuit, and a detection. You must select and justify suitable operating parameters for the circuit and may implement additional enhancements (general features).

Your team must include two pairs of probe points to allow the **voltage** across: (a) the trigger pin of the 555 timer and (b) the output pin of the 555 timer to be measured as part of the assessment of this subsystem. For more information on probe points, please review the component guide section of the project guide.

The following items comprise of the mandatory components to be implemented for this subsystem.

Category	Description	M3 Points
Alarm Circuit	An astable 555 timer and buzzer are used to generate an alert tone between 300 and 1000 Hz. The alert tone should sound immediately upon detection of a power outage.	5
Alert Tone	The volume of the alert tone is set to be clearly audible, but not excessive.	2
System Operation	The entire failure alert system, and all options, must function independently from the Arduino/integrated system. Hint: This means it must not draw power from or depend on the Arduino's power pins.	3
Detection Circuit	Your alert system should detect a loss of power to the Arduino via monitoring the Arduino's power pins.	3
Alarm Circuit	The frequency of the tone generated by the astable 555 timer is adjustable in hardware over at least two options.	2

The following items comprise of the optional components to be implemented for this subsystem.

A maximum of <u>10 points</u> selected from the optional items (in orange) from this subsystem will contribute to your M3 points total. You can take on more than the maximum points in value from this subsystem but only the maximum will apply to the total. **Features that require a choice of implementation are highlighted in a darker colour.**

Category	Description	M3 Points
Alert Tone	A two part-tone is utilised using additional circuitry. Each of the tone components must be sounded for at least 0.25 s in sequence, with at least 200 Hz between the components.	5
Detection Circuit	Your alert system maintains the alert tone for a period of 5 s after power is restored, after a power outage has occurred. Note: this means that the alert tone should continue for 5 seconds once power is restored after a power outage. This feature should not impact the start of the alert tone described in the Mandatory Alarm Circuit above.	5
Adjusting Frequency	A user can adjust the frequency of the buzzer in hardware (e.g. with a switch, knob, or similar): ● Option 1: over a small continuous range of values (at least +/- 100Hz); ● Option 2: over a large continuous range of values (at least +/-400Hz); ● Option 3: both tones (requires two-part tone feature) may be tuned over a small continuous range of values (at least +/- 100Hz) Note that physically removing or changing a component will not be considered a successful demonstration of this feature.	3 Option 1 5 Option 2 5 Option 3
Duty Cycle	The duty cycle of all astable 555 timers must be an appropriate, selected value and uniform across all timers.	2

4.2.5 Integration

System integration requires the subsystems to interact with each other. Therefore, your team may only use a single Arduino for your integrated system. 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 or an R-2R ladder arrangement to manage pin usage.

The maximum number of points you can earn without final integration is 85% of the total value of this milestone. You will always earn more points for integrating additional subsystems. The mark you will achieve for integration will depend on how many of the subsystems run off a single Arduino on the day, for example if you run 3 subsystems off one Arduino you will achieve 2 points.

However, 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** additional integration, as **trying to do so at the "last minute" will often result in accidental breakages to your system**.

Description	Points
Three subsystems integrated.	2
Four subsystems integrated.	5
Five subsystems integrated (must have working FA subsystem).	15

4.2.6 Milestone 3

Milestone Deliverables:

The full functional system will need to be demonstrated so that its functionality can be tested. This will include the control, service, input and output subsystems as specified in the project brief under MVP, and features as selected under FP. This Milestone will include Moodle submissions and an in class presentation of the built prototype (more information on testing of the prototype is under the testing subheading).

By the Milestone 3 due date on Moodle the Circuit Diagrams for all the features you intend to demonstrate as a single PDF file (such that each circuit has its own page) and Python Code written to implement all the features you intend to demonstrate must be submitted. Files should be sensibly named .

Teams may choose to include a short video demonstrating the functionality being assessed in their moodle submission. This will serve as a backup for assessment if the demonstration during class fails (see deliverables/section 4 for details on how this will be marked).

Note that if your team does not have an approved M3 feature-set via the SYSTEM by the deadline (Tue 11:55pm Campus Time W11), your team will be marked as if you selected all items.

Points caps per subsystem for M3:

Points caps are applied to each subsystem - you can pick more then the cap amount of points but you will at most earn that cap amount from the subsystem. It is strongly recommended for students to pick more than the cap in order to cover for where they might lose some marks. Please note that deductions will only be applied after the total points earned is calculated.

- Services Subsystem: 15 points
- Inputs Subsystem: 15 points
- Outputs Subsystem: 30 points
- Failure Alert Optional: 10 points
- A cap of 70 points will be applied to points calculated from the four items above.

Milestone Weighting:

In each specification the mark for the item in relation to Milestone 3 is listed as M3 points. This is out for a total of 100 points.

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).

Submission Format:

For full submission marks your code submission should meet the coding standards provided. The circuit diagrams should be titled and should be one circuit per page, with each page being A4 (or larger as necessary) in size, make sure that your circuit diagrams are readable.

Your folder may contain additional .mp4 videos demonstrating functionality as required.

Put your project code and circuit diagrams in a folder named TeamNNN, replacing NNN with the three character for your team number, i.e. TeamG05. Then ZIP the folder up and name it TeamNNN-M3.zip, replacing NNN with the three character for your team number, i.e. TeamG05-M3.zip

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.

4.2.6.1 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). For more information on probe points, please review the component guide section of the project guide. Discuss the implementation requirements with your manager if you have any questions.

Category	Description	Deduction
General	Console print statements are unclear, have insufficient information, or are poorly formatted.	-2
	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.	-3
	The code cannot return to the main menu from anywhere in the polling loop during normal operations.	-3
	Code crashes, does not exit cleanly, has an unhandled exception, uses unspecific exception handling, or exits with an error message.	-2
	The seven segment display current or voltage is too high for the display segments.	-2
	Probe points (terminal header + jumper header pins) are not included for measuring current and voltage on the seven-segment display.	-2
	Probe point (terminal header + jumper header pins) is not included for measuring voltage on the failure alert subsystem timer ICs output pins.	-2
	Probe point (terminal header + jumper header pins) is not included for measuring voltage on the failure alert subsystem timer ICs trigger pins.	-2
Circuit	Circuit Diagram components are in the wrong orientation, power source and ground not oriented correctly (source at top left, ground bottom right), labels not oriented horizontally and readable.	-6
	Circuit Diagram doesn't use clear, well labelled nodes (including missing labels), or are labelled incorrectly, or is missing the title block, or doesn't use the correct component diagram	-7
	Physical circuit doesn't follow wire colour conventions, doesn't use rails appropriately, or uses spaghetti wiring.	-2
Coding	TODOs left in code	-2
	Variables not in lowerCamelCase	-2
	Functions not in snake_case	-2
	Indentation not 4sp = 1 tab	-2
	Function header documentation in wrong format, incomplete, missing or not present	-2
	File header documentation in wrong format, incomplete, missing or not present	-2
	Non permitted imports exist, or non permitted techniques used.	-10 ea

Good Practice Requirements

Following good practice for code and circuit diagrams is essential for the practice of engineering in industry. Therefore, you must follow the Circuit Diagram Standards and Coding Standards as specified by this unit. Deductions will be made for submissions that do not conform to these specifications, as well as for incomplete submissions, or for using hardware components and software packages that are not permitted by these specifications. **Good practice guidelines will be assessed and must be adhered to.**

Further, please note that we will not assess any features not submitted in code/circuits on Moodle by the Friday night milestone deadline in your practical class. Therefore please ensure you review your submission carefully.

Description of Deductions	Points
Circuit diagrams missing for a given function demonstrated as part of Milestone 2 or 3.	100% of points earned for that feature in the milestone.
Circuit diagrams deviate from the quality and standardisation specifications listed in the Circuit Diagram Standards document.	up to 10% of the team's milestone store will be deducted.
No circuit diagram submitted.	Maximum penalty will be taken from circuit standards.
Code missing for a given function demonstrated as part of Milestone 2 or 3.	100% of points earned for that feature in the milestone.
Code does not meet all quality metrics listed in the Coding Standards document (including use of classes and assert statements)	See deductions in standards above. Use of classes and assert statements will result in a 10 point penalty, followed by remarking with the offending statements commented out.
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.
Other packages besides the allowed software packages are used (see list in section 2).	See deductions in standards above. The offending package import will be commented out and marked as is, with a 10 point penalty.
Individual team member absent from demonstration.	**No points will be awarded in** Milestone 2 or 3 for the **individual member** if special consideration **is not** granted.

4.2.6.2 Demonstration

To confirm functionality of the designed prototype traffic system the Milestone 3 testing will be performed using the model traffic system specified in section 3. You will have access to this system in the later half of semester to confirm calibration and functionality of your developed system.

Each group will be allocated a time slot within the 3 hour practical session in which they will need to demonstrate their prototype using the model. We will test each of the features listed in your M3 feature submission (last approved version).

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.

All of your team must be present at the presentation. **If you are late or absent, you will not receive marks.**

4.2.6.3 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 a reduction as the 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

4.3 ITP Metrics Peer Assessment

In Weeks 4, 8 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.

After the final peer assessment in Week 12, the **system will generate a "Peer Assessment Factor" (PAF),** 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.

4.4 System Extension

Whilst you have completed your prototype there are a range of ways which the system can be utilised in the field using a range of other technologies. The final section for the project will be a discussion based on any possible developments. **No code or hardware will need to be submitted**, as this discussion is purely theoretical.

4.4.1 Milestone 4

Milestone 4 is a team reflection on the entire project, its limitations, and potential extensions to it. You should discuss:

- the implementation of your selected features and your systems integration strategy,
- possible uses of machine learning as an extension of the features of your smart system,
- the data ethics implications raised by the data collected by your system.

4.4.1.1 Part 1 - Video

As a team, you must create a 10 min video discussing the topics listed above. This video should be in MP4 format and named as TeamNNN-M4.mp4, where NNN is the team number - i.e. MG05. You can use zoom or other video creation software but the video should not be full HD - max 720p.

Description	**M4 Points**
Implementation and integration	5
Machine learning	5
Data ethics	5

4.4.1.2 Part 2 - Peer Comments

As an individual assessment item, you will receive 15 points for leaving helpful and relevant comments of at least total 100 words on three (3) other team's videos on the Moodle forums after the primary submission of this milestone. You will need to fill out the review checklist by answering the statements and either agreeing or disagreeing with the statements. You will be allocated 3 or more videos - you should review at least 3.

Description	**M4 Points**
Useful, relevant commentary on three (3) other team's submissions (individually assessed)	5 x 3