MECHENG 201

VEX Project – Part 2

Multiple due dates (worth 20%)

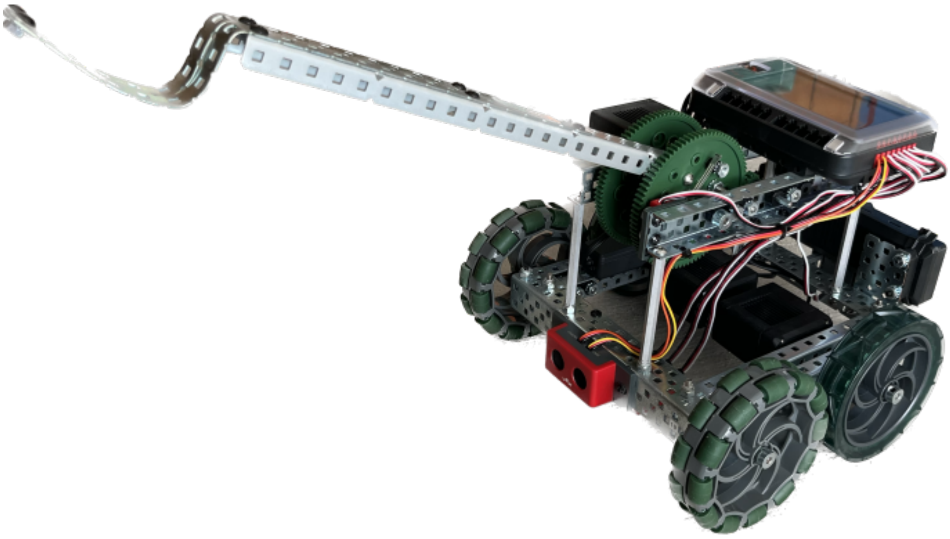
Name:

Project Partner:

Project Partner's Mobile Number:

Lab Stream:

Robot #:



Semester 1, 2024

Contents

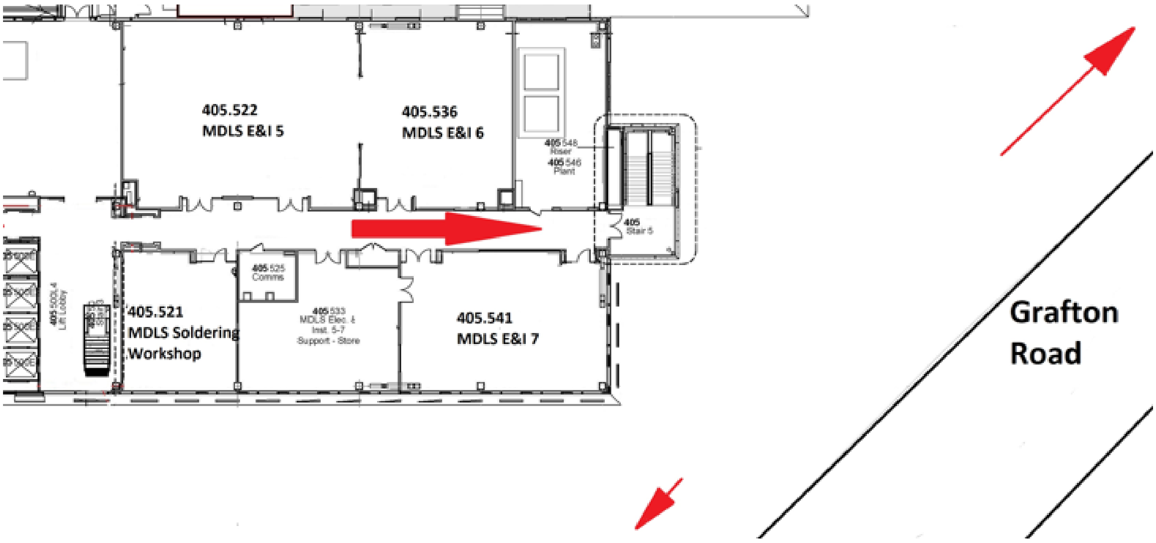
Safety Induction for B405-536 MDLS Electrical & Instrumentation 6

Welcome to MDLS E&I 6. This information has been provided to you because we are genuinely concerned about the health and safety of everyone here. As a University and an engineering student, we ask that you:

- Be properly attired – no Sandals, no jandals, no bare feet, or loosely fitting clothes.
- Secure long hair when approaching or working close to the machinery. Jewellery must be removed as with any loose items.
- Maintain proper decorum at all times.
- Be conscious of personal safety and of those working nearby at all times.
- Clean up equipment, tools, work area and surrounding floor after use.
- Stow away items, tables and chairs neatly at the end of the session.
- Report all injuries (including minor burns, bruises and cuts) immediately.
- Be familiar with the location of the first aid kits.
- Comply with the emergency Procedures – Note exit routes, location of phones, fire extinguishers.

Evacuation Plan

In the unlikely event of an emergency and immediate evacuation is initiated, please note the closest EXIT door of the learning space. The evacuation routes are shown (RED arrows) in the plan. The doors will be marked with a green coloured and lighted EXIT sign.



1 Learning Objectives

This project will introduce you to the basic principles of computer-controlled devices that can change a mechanical output based on sensor measurements from the environment. You will gain an appreciation for programming that deals with physical uncertainties. By completing Part 2 of the project, you will

- Be able to write functions that precisely control the movements of the VEX robot using closed-loop control techniques.
- Be able to write code that follows a specific line on the floor.
- Be able to implement and tune P and PI controllers.
- Find and understand the physical limitations of the the robot and closed-loop control.
- Draw complete flowcharts that describe the logic of your programming solution.

This part of the project is worth 20% of your final grade. You will need to demonstrate your solution using the physical robot during your ***only*** lab session in week 9. The source code, project report, and peer assessments are to be submitted electronically on Canvas on different due dates in week 10.

2 Background and Task Description

Due to the rising cost of labour, you are contracted by the Dairy Industry Cooperative to develop an autonomous robotic solution for delivering stock/payload (weighted cans). They were pleased with your autonomous solution in their small storage room (Part 1 of the project), and now they want to commission you to develop an autonomous solution for their large warehouse facility (Part 2 of the project). A scaled down robot has been provided to you, along with a map (Figure 1) that represents features in the real world.

2.1 The Task

In the warehouse facility, the automated forklift (VEX robot) is required to leave it's charging station indicated by *Start Area* (green hatched area) in Figure 1. The objective is to pick up the fragile payload (weighted can) from the predetermined location (anywhere between the yellow lines) as shown in Figure 1, and drop it off as accurately as possible on the designated drop-off zone before returning to the charging station (finish area – yellow box in any orientation). The robot wheels must not leave the designated area bound by the warehouse walls (board outline) and stock shelves (red hatched areas), as doing so will risk damaging the property or equipment. Additionally, there is a 120 second time limit, as the real robot will run out of (scaled) battery charge, and the production chain will need to be halted to retrieve the robot. Refer to Section 3 for the complete set of rules. Also refer to Section 6 for useful hints and cautions.

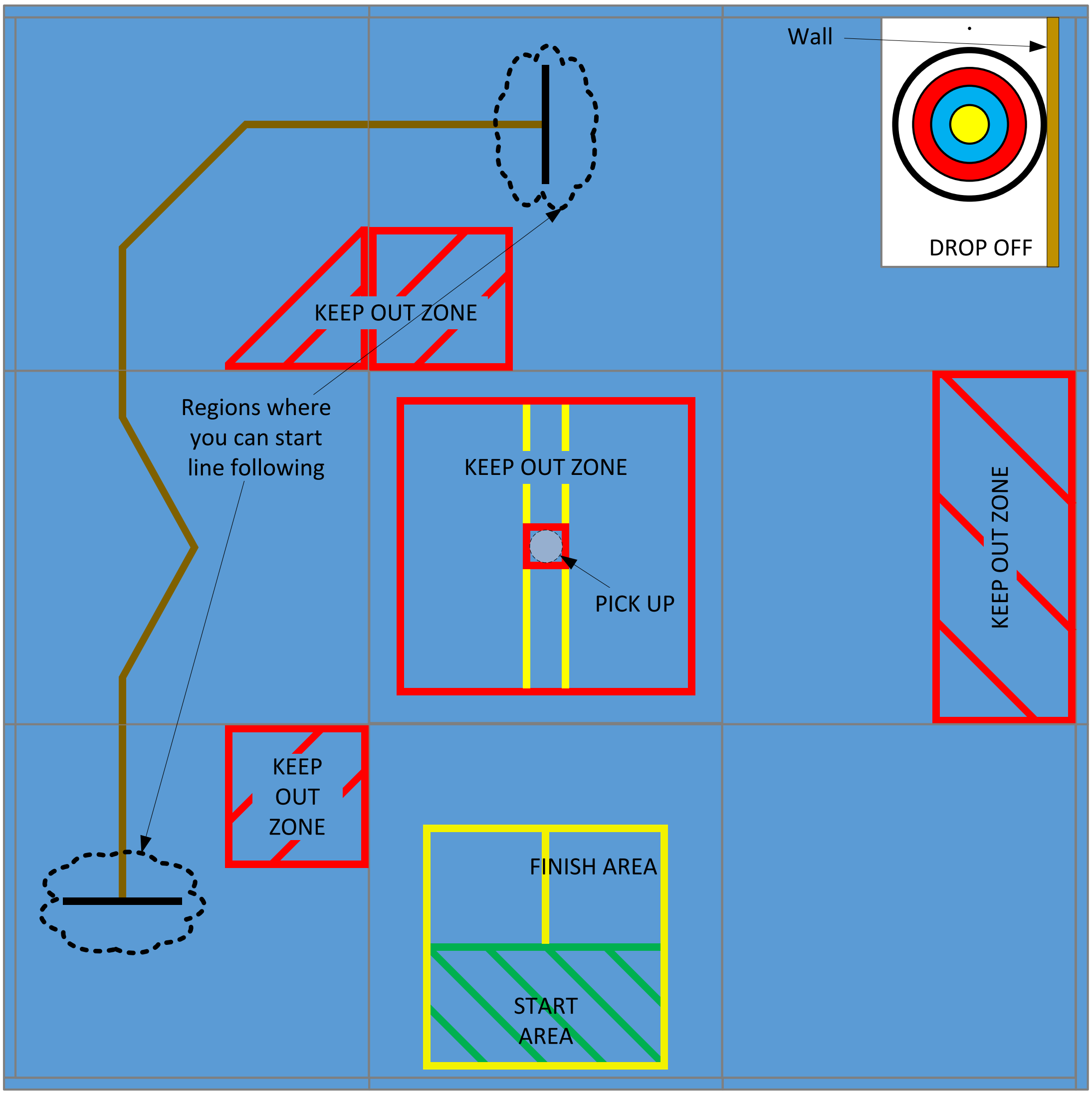


Figure 1: Warehouse layout.

2.2 Warehouse Layout

There are three ways in which the robot can navigate around the map.

2.2.1 Encoder-Based Navigation

In this part of the map the robot must navigate around the keep out zones (red-coloured areas) using the encoders to determine its position or how far it has moved/turned. Therefore, you must create a function to accurately drive a specified distance and another function to turn a specified angle. Appendix A provides important dimensions for the map and the VEX robot.

2.2.2 Line Following

Line following uses the light sensors to follow the brown line on the map. Black stop lines perpendicular to the brown line indicate the start/end of that line segment. In this part of the map, you cannot use the encoders to turn a specified angle or drive a certain distance. Encoders can only be used to ensure the robot is driving straight.

2.2.3 Ultrasonic-Based Navigation

For driving to pick up the payload, the robot must use its ultrasonic sensor to measure the distance to the payload. Therefore, you must create a function to accurately drive the robot to be within a specified distance to an object in front of it.

3 Rules

1. All four wheels of the robot must start inside the start area and finish inside the finish area. The robot can finish in any orientation inside the finish area.
2. No further contact with the robot or payload is permitted once the program has started.
3. The robot must start and end with the arm at its highest position.
   (a) For safety reasons, the arm must be raised all the way up when turning.
4. The robot must perform a loop. That is, it cannot return along the same path.
5. The task should be completed within 120 seconds, with points deducted for exceeding that time limit by up to 20 seconds, and disqualification of the run thereafter.
6. The robot must stop for 1 second at the black line before starting the line following segment.
7. The arm and the wheels cannot be moving at the same time.
8. You cannot use the encoders to turn a specified angle or drive a certain distance while the robot is following the line.
9. Your entire project code must be in a single file (plus the header file).
10. Your run will be disqualified if:
    (a) Any of the above rules are violated,
    (b) the robot fails to pick up the payload,
    (c) the robot drops the payload or the payload slides down the robot arm,
    (d) all four wheels of the robot leave board area or all four wheels enter a keep-out zone,
    (e) any wheel remains outside the board or inside the keep out zone for more than 2 seconds,
    (f) the robot tips over,
    (g) the robot does not return to the finish area, or
    (h) a human interacts with the robot while it's moving.

4 Demonstration and Scoring

You will be given a total of ten minutes (split into two five-minute assessments) to setup and demonstrate that your solution works repeatedly. Your final score will be the average of your best two runs across the entire ten minutes. If you only manage to complete a single run in 10 minutes, your second attempt will be given a score of 0 and therefore your final score will be half of your only run.

4.1 Scoring

A maximum score of 100 points is possible for each run, and is comprised of following elements:

4.1.1 Robot Manoeuvring (40 points)

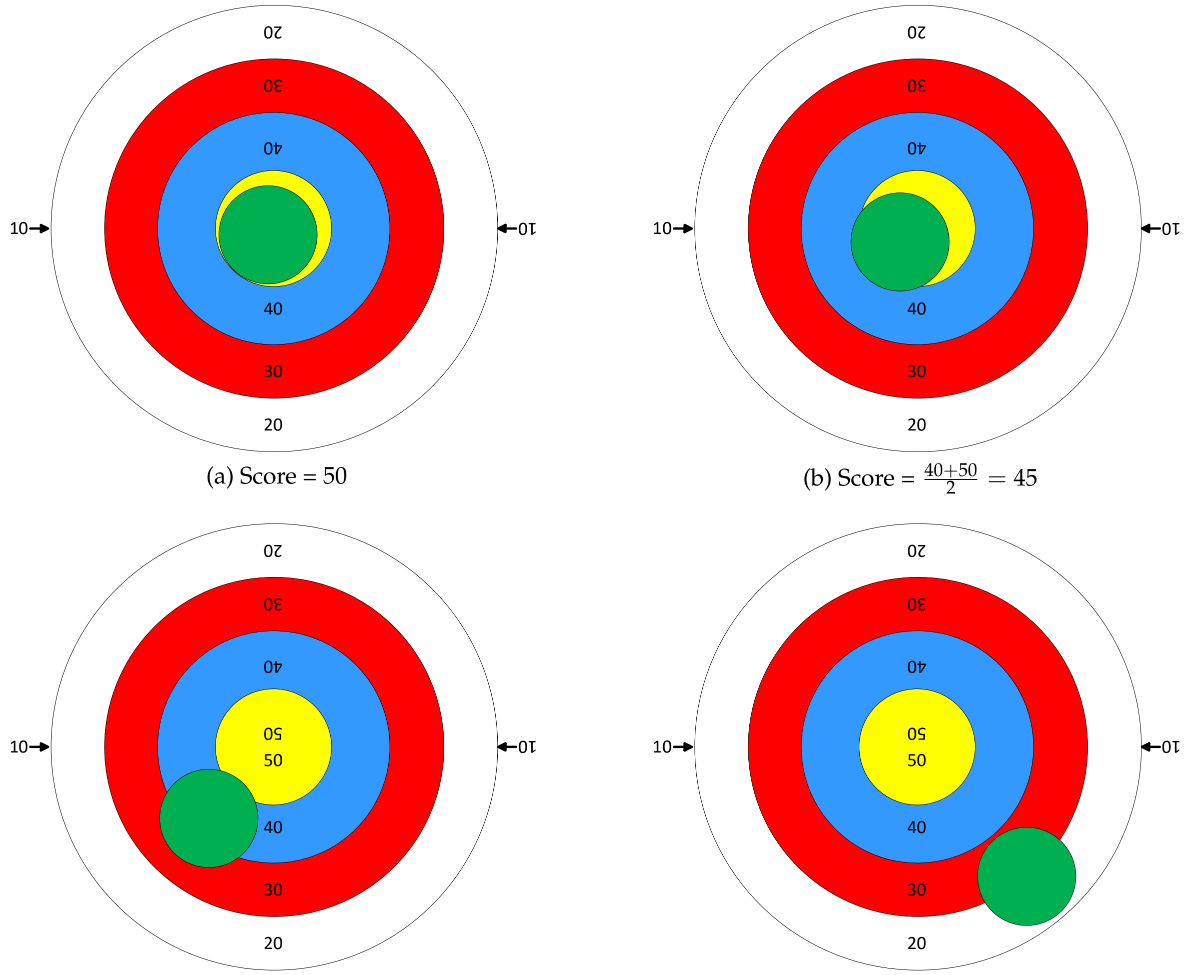
The robot manoeuvring tasks listed below need to be demonstrated only once to get the points associated with them. These points will be added to your base score (up to 40 points) of each run.

- 10 points for lowering the arm to an appropriate height.
- 10 points for successfully lifting the payload using the ultrasonic sensor.
- 5 points for basic line following.
- 5 points are awarded if the robot successfully completes the line following segment.
- 5 points for basic encoder-based navigation.
- 5 points are awarded if the robot is driving straight with smooth starting and stopping in the encoder-based navigation part of the map.

4.1.2 Placement Accuracy and Returning Home (60 points)

Unlike the robot manoeuvring points, the 60 points below are calculated and recorded after each run.

- Up to 50 points measured by the average of all the different scoring areas the base of the payload is touching (see Figure 2). If the base of the payload lies completely outside the target, a score of zero will be given.
- 10 points for returning to the finish area (home) in under the time limit of 120 seconds after placing the payload in the drop-off zone.



(a) Score = 50

(b) Score = $\frac{40+50}{2} = 45$

(c) Score = $\frac{30+40+50}{3} = 40$

(d) Score = $\frac{30+20}{2} = 25$

Figure 2: Load placement scoring.

4.2 Penalties

Penalties will applied if any of the items below occur. A maximum penalty of 20 points can be given.

- Going out of bounds or entering a keep out zone will incur a penalty of 2 points per wheel for each incident.
- 0.5 points deducted for every second over the time limit of 120 seconds.
- 2 points for each wheel left outside of the finish area at the end of the run.

4.3 Time bonus

If the robot completes its run in less than 60 seconds, 10 points will be awarded after any penalties have been applied. The time bonus is only awarded if the payload is placed onto the drop-off zone and the robot has returned to the start/finish area. Note that the final total score cannot exceed 100 points. This bonus cannot be transferred to the project report mark.

4.4 Disqualification

If the run is disqualified, then you'll get whatever score you accumulated up to that point. For example, if the robot used line following to place the payload onto the drop-off zone before being disqualified, then you'll get 10 points for positioning the arm, 10 points for lifting the payload, 10 points for line following, and the score for placement accuracy as well as any accumulated penalties up to that stage.

5 Program Requirements

To help you write a structured and logical program, your program must contain and *use* the following:

- A function that drives straight a user-specified distance in millimetres and then stops. Negative distances mean reverse. This function must use a PI controller for controlling the distance travelled. The control error must be in millimetres.
- A function that drives straight to be a user-specified distance in millimetres away from an obstacle (using the ultrasonic sensor). This function must use the ultrasonic sensor to measure how far the object is from the robot, calculate the required distance to travel, and call your function to drive a specified distance. This function must not move the arm.
- A function the takes wheel encoder counts and converts them to travelled distance in millimetres.
- A function that rotates the robot a user-specified angle in degrees and then stops. Positive values correspond to counter-clockwise rotation and vice-versa. It is recommended that the robot rotates about its central pivot point. This function must use a P or PI controller. The control error must be in degrees.
- A function that rotates the arm to a specified angle in degrees. Zero degrees corresponds to the arm in the horizontal position. This function must use a P or PI controller. The control error must be in degrees.
- A function that takes arm encoder count and converts it to the position of the arm in degrees.
- A function the drives straight at the given power level until any of the sensors detect black line. It must use the wheel encoders to ensure it's driving straight.
- You must saturate all motor commands before sending power to the motors.
- The commanded control efforts, u and Δu, of your controllers must be in percentage power. You must also use your `convertPower()` function from Part 1 of the project.

You are encouraged to write additional functions and tasks that simplify your code and/or improve its robustness, efficiency, or accuracy.

6 Hints and Cautions

- Start by drawing flowcharts of your overall program and of any other major function. The lab teaching assistants (TAs) have been instructed not to help unless you can show them a flowchart of your program/function you are seeking help.
- Use the template folder on Canvas as a starting point. Improve the functions you created in Part 1 of the project and use them in this part of the project.
- Remember that 50% of the demonstration marks are on placement accuracy. So, focus on making a working program first before going for speed.
- If you have a good line following function, use that to get to the drop off zone as it will be more consistent/accurate. Otherwise, use the encoder-based navigation part to get to the drop-off area and use line following to return to the finish area to maximise your score.
- The same light sensor reading the same colour on the same surface can vary, and different light sensors can return completely different values for the same colour on the same surface.
- Below a certain angle, the robot arm will start to interfere with the ultrasonic sensor readings. So, it's best to read the sensor at the start when the arm is up, and use your distance driving function to get to the payload.
- If your code does not work as expected, check that the sensors and motors are functioning before investing time to debug your code.
- The nuts, bolts, and stand-offs can occasionally come loose. Notify the lab TAs if that happens. It is your responsibility to make sure everything is tightened and stable before your test runs.
- There is no guarantee that you will have a fresh battery at the beginning of each lab session. Check the battery charge at the start of each lab session. If it is below 10%, let us know.
- The wheel motors may not perform exactly the same (i.e. not driving straight) and you may need to account for this by implementing a control loop.
- Test sequentially – if you have a long program that requires previous steps to work before moving on to each new step, make sure the previous steps work first.
- The robot may drive differently on different surfaces. You can test on the floor, but to make sure that it will perform as expected on demonstration day, you should also test it on the actual boards.
- Be consistent with your starting position and how you setup the robot. A small change in the starting position can lead to a completely different outcome.
- Due to **backlash** in the wheel motors, **you will always have uncertainty and randomness in each run**. Test multiple times before committing to change in the code.

7 Assessment

The project demonstration will be conducted during your only lab session in week 9. Demonstrations will be held in 405-536. The final project mark will be based on the following:

7.1 Week 6 Checkpoint (4%)

During your second lab session in week 6 of the semester, you need to demonstrate that you can do at least four of the following:

- [Highly recommended] Driving a specified distance using a PI controller
- [Highly recommended] Turning the robot a specified angle using P or PI controller
- [Highly recommended] Following the brown line until it detects a black line.
- Controlling the arm angle using a P or PI controller.
- Driving until the robot is a specified distance away from an obstacle.
- Driving straight until the light sensors see a black line.

Each of the above tasks is worth 1% (up to 4% in total).

7.2 Robot performance (36%)

A summary of all scoring possibilities is listed in Table 3 below. Refer to section 4.1 on page 5 for more details.

Table 1: Robot performance scoring breakdown.

Score Component	Points
Robot manoeuvring	
Arm control	10
Reaching and lifting the payload	10
Line following	10
Encoder-based navigation	10
Placement accuracy	50
Returning to the start/finish area	10
Penalties	up to -20
Time bonus	up to +10
Total (rounded to nearest full point)	100

7.3 Report and Code (60%)

Your report and code will be marked independently from your demonstration and according to the categories listed in Table 2.

Table 2: Breakdown of the report marks.

Category	**Marks**
Clear and well commented code	10
Program logic and flowcharts:	30
- Correct logic for completing the task	
- Correct use of flowchart symbols and connections	
- Code matches flowcharts	
Report quality (well written, formatted, and structured)	20
Program quality	40
Total	100

For the program quality category, higher marks will be awarded to a ***robust and efficient*** program that requires minimal calibration (i.e. a program that works on any robot with little or no calibration), contains internal checks, and contains code to overcome some physical/robot uncertainties. Marks will be deducted for inefficient or bad programming practices (e.g. using open-loop control for robot manoeuvring, poor choice of variable names, incorrect code layout and indenting, repeated functions with minor changes only such as a function to move forward and another to reverse, etc.).

7.3.1 Code

Your code will be submitted separately from the report. Do not include the code in your report. Refer to Section 8 for more details on the submission requirements.

7.3.2 Report

A brief formal report (maximum 6 pages long excluding appendices and title page; minimum font size is 12, 2 cm page margins all around) should contain:

- A brief introduction to the project and task.
- Description of your solution. This should explain the **logic** of the program in English with the aid of flowcharts and diagrams, and point out any ***special features*** that your code has. You **must** include **TWO flowcharts** in the main body of the report: one for your main program and one for your line-following algorithm. Additional flowcharts for subroutines can be included in the appendices if you think it will make it easier for the reader to understand your code logic.

  - The flowcharts must match your code.
  - The flowcharts can be drawn neatly by hand, scanned, and inserted into your report.
  - The flowcharts should not contain code or function names. They should be in English and can contain mathematical expressions.
- A section reflecting on how the robot performed on the demonstration day and your recommendations, if any, on how the performance could be improved.

8 Submission Deadlines and Requirements

There are two group submissions and two individual submissions throughout the duration of this part of the project.

8.1 Group Submissions

For group submissions, only one person from the group needs to submit. The other group member(s) ***must*** double check the submission made *before the deadline*. Late submission penalties will apply for the whole group.

8.1.1 Source Code

Your source code (`*.c` `*.h` and files) and a PDF printout of your code need to be submitted electronically as a group on **Canvas** no later than **11:59 pm on the day of your demonstration in week 9**. The code you submit must exactly match the code printout. The printout of your **commented** copy of the code must include **line numbers**. Each function you have written should have a short description indicating its purpose.

- Do not print your code directly from VS Code. Instead, **open your files in Notepad++** and make sure to enable the option to add line numbers when printing. Print to a PDF printer.

8.1.2 Report

The group report is to be submitted **electronically by 11:59 pm on Monday 13 May on Canvas**. Make sure the title page includes your group name (e.g. E07). The report must be uploaded in a **SINGLE PDF** file format. Failure to follow this instruction will incur additional penalties.

8.2 Individual Submissions

All members of the group must contribute equally and fairly towards all aspects of the project (programming, drawing flowcharts, report writing, etc.). There will be two peer assessments (PA) to be completed for this project. Each member of the group needs to make their own submit. Late submission penalties will apply to the individual student and not affect the group.

8.2.1 Preliminary Peer Assessment

In week 6 of the semester, you are required to complete a simple online peer assessment by **11:59 pm Thursday 25 March**. In this peer assessment reflect on the contributions made in parts 1 and 2 of the project thus far. Failure to complete this PA may result in a 5% individual penalty applied to your project score. The purpose of this PA is to identify any issues early on and address them before it is too late.

8.2.2 Final Peer Assessment

Each member is required to fill out a confidential peer assessment form and upload it to Canvas. The form (Compulsory Peer Assessment.xlsx) can be downloaded from Canvas closer to the submission date of the project. A late PA submission will incur a late submission penalty applied individually (i.e. not affecting your partner's mark). The peer assessment must be submitted by **11:59 pm on Tuesday 14 May on Canvas**.

8.3 File Naming Requirements

Please use the file naming convention below for naming your files.

- Your lab group designation is based on your lab stream and robot number.
- For the report PDF, use `yourLabGroup_YourSurname_PartnerSurname.pdf`
- For the peer assessment file, use `yourLabGroup_YourSurname_PA.xlsx`

For example, let's assume your surname is Smith and you are in group 7 of lab stream E and your partner's surname is Doe. Then, your group name is E07 and your files will have to be named as follows:

- Group submissions:
  - Code file remain unchanged: `Student_Code.c` and `Student_Code.h`
  - Report PDF: `E07_Smith_Doe.pdf`
- Individual submission – Final Peer Assessment: `E07_Smith_PA.xlsx`

You can resubmit your files if you made a mistake but note that we will only consider your latest submission. When resubmitting, you need to upload all the necessary files for that submission again. Resubmitting after the deadline will incur late submission penalties. If you resubmit your files, Canvas will automatically append -1 or -2 to your file names. That's fine but the original file name should be the correct one.

8.4 Late Submissions

Canvas automatically flags late submissions if they are submitted after the deadline even if your submission was uploaded a few seconds after the due date. Please make sure to submit at least 5 minutes before the deadline to avoid any late submission penalties.

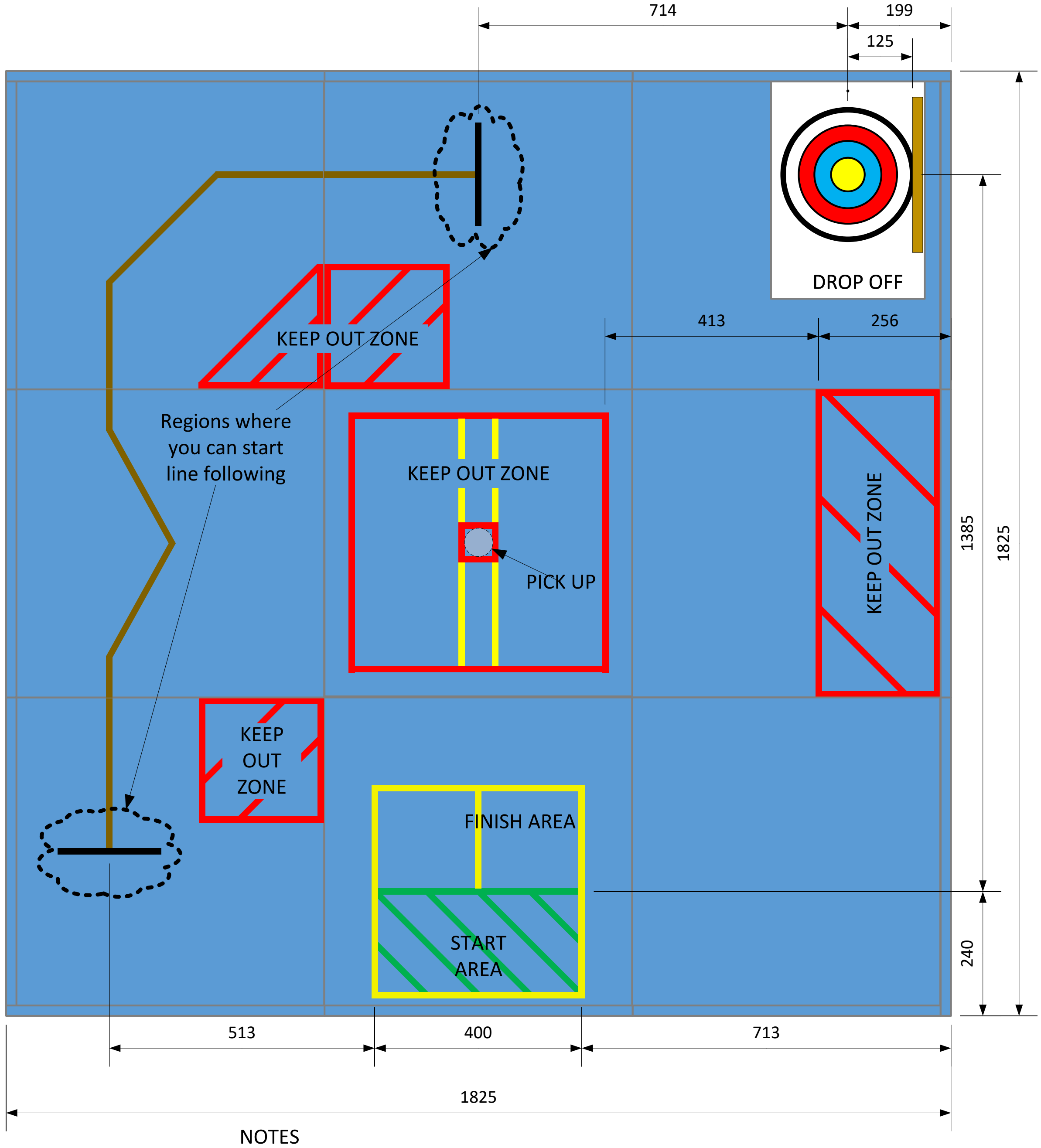
The penalties listed in Table 3 will be applied for late group submission (for group penalty) and PA submission (individual penalty):

Table 3: Late submission penalties

How late?	Penalty
0 to 30 minutes late	5%
31 to 60 minutes late	10%
1 hour to 12 hours late	30%
12 hours to 1 day late	50%
More than 1 day late	100%

Appendix A Map Details and Robot Dimensions

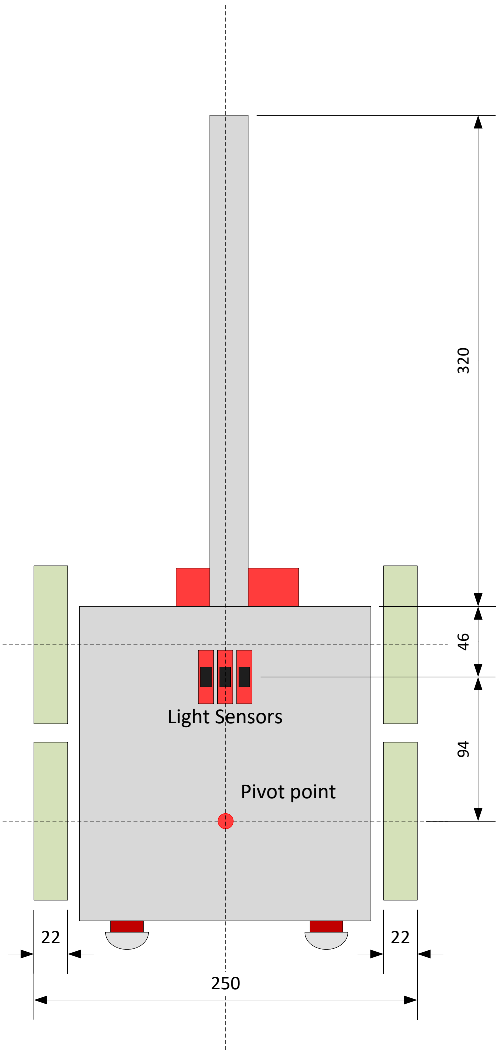
The key dimensions for the scaled down warehouse facility and robot dimensions are shown in Figures 3 and 4, respectively.



NOTES

- All dimensions in mm
- All dimensions are approximates and measured from line centres
- General tolerance of ±5mm
- No extra dimensions will be provided

Figure 3: Important map dimensions.



Notes
- All dimensions in mm
- These are approximate dimensions
- Pivot point identified assuming both wheels turn at the same rate
- The arm and middle light sensor are not exactly at the centreline of the robot
- The distance to the tip of the arm is measured when the arm is approximately horizontal (parallel to the ground).

Figure 4: Important robot dimensions.