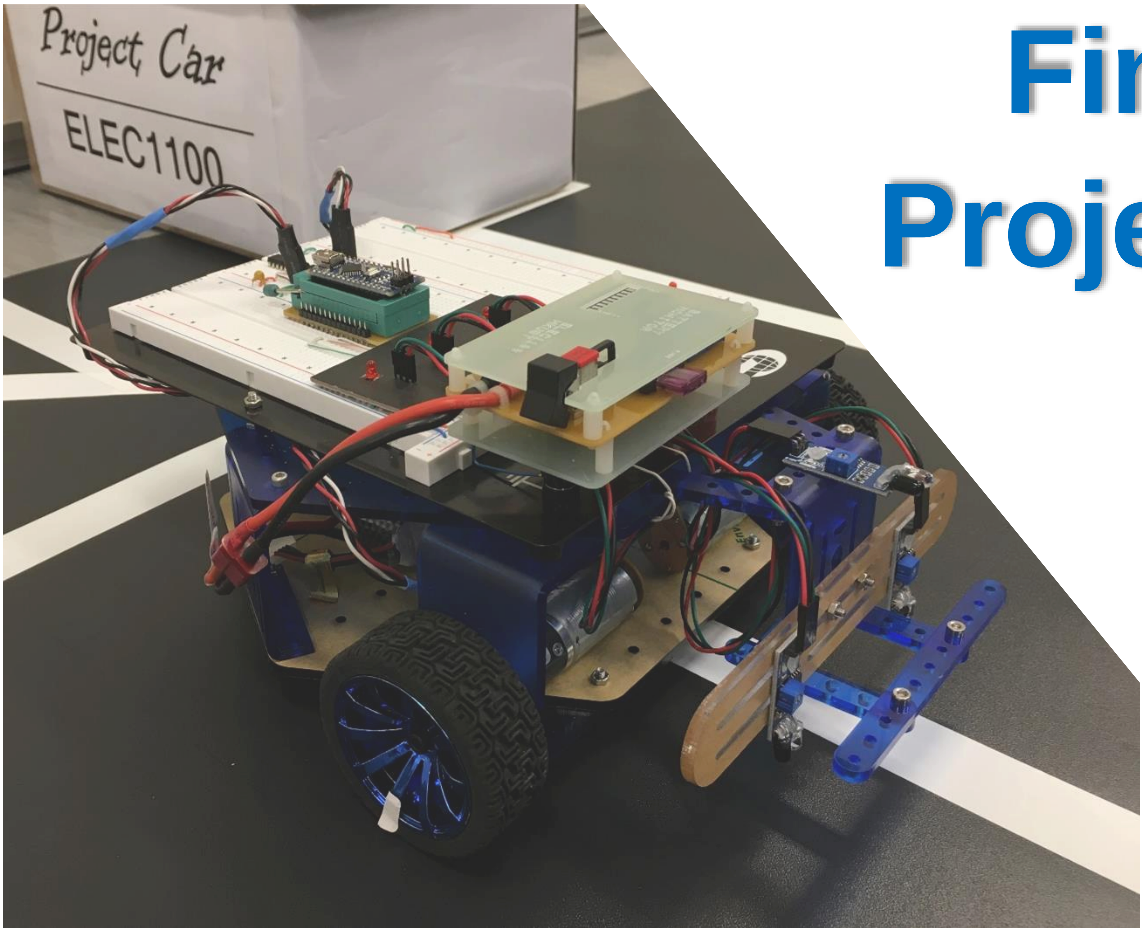
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY
Department of Electronic and Computer Engineering

ELEC1100 Final Project



Contents

This project accounts for <u>25%</u> of your overall grade: demo (20%) and report (5%).

A) Objectives

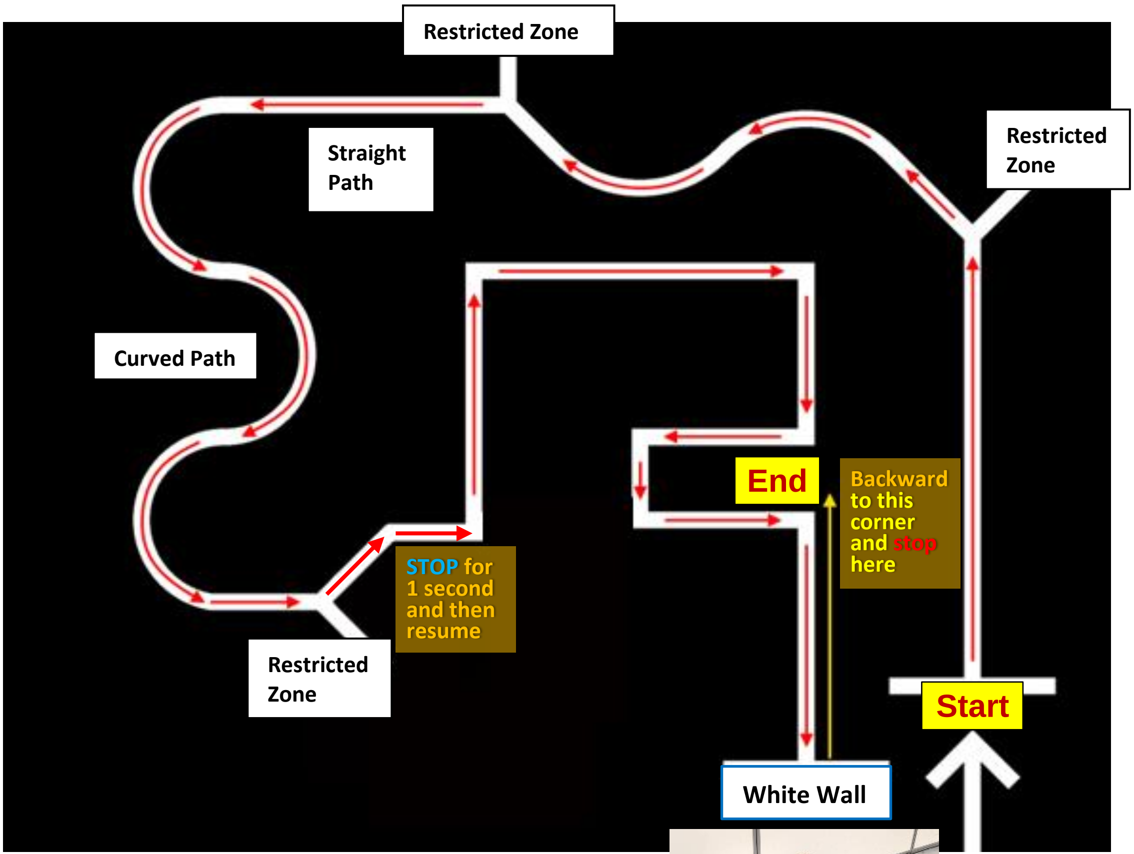
- Put together your work from Lab#03 to Lab#06 to design a vehicle that can navigate through an obstacle course.
- Write a final report to summarize your work in logic design of controlling the vehicle.

B) Introduction

By now, you have learned many important concepts and techniques needed to build an autonomous vehicle. For the final project, you will optimize your design and enrich your Lab#06 code to tackle the final design challenge.

To allow the project to be completed in an online mode, but provide a real hardware experience, we will download and run your Arduino code on a real robot car and do tracking to determine how well your logic system can operate. Please read the following instructions thoroughly to understand the details of how to do this.

In the final project, a robot car is required to move on an obstacle course from **Start** to **End** (See **<u>Part E</u>** the "Grading Policy for the Track Demo"). The course is a white line on a black surface with several challenging stretches and a white wall at the end (a box with white paper on front). Below is a picture of the final course. The proportions and dimensions may not be on scale but it should give you an idea of what the real course looks like.



C) Rules for the Final Project

- About your vehicle:
  The **robot car** will be provided by the teaching team at the test & demo sessions, configured with completed control circuits on the breadboard including an Arduino-Nano board (see Photo 1) and 3 line sensors (bright sensing) at working positions (see Photo 2).

Photo 1: breadboard layout of your project

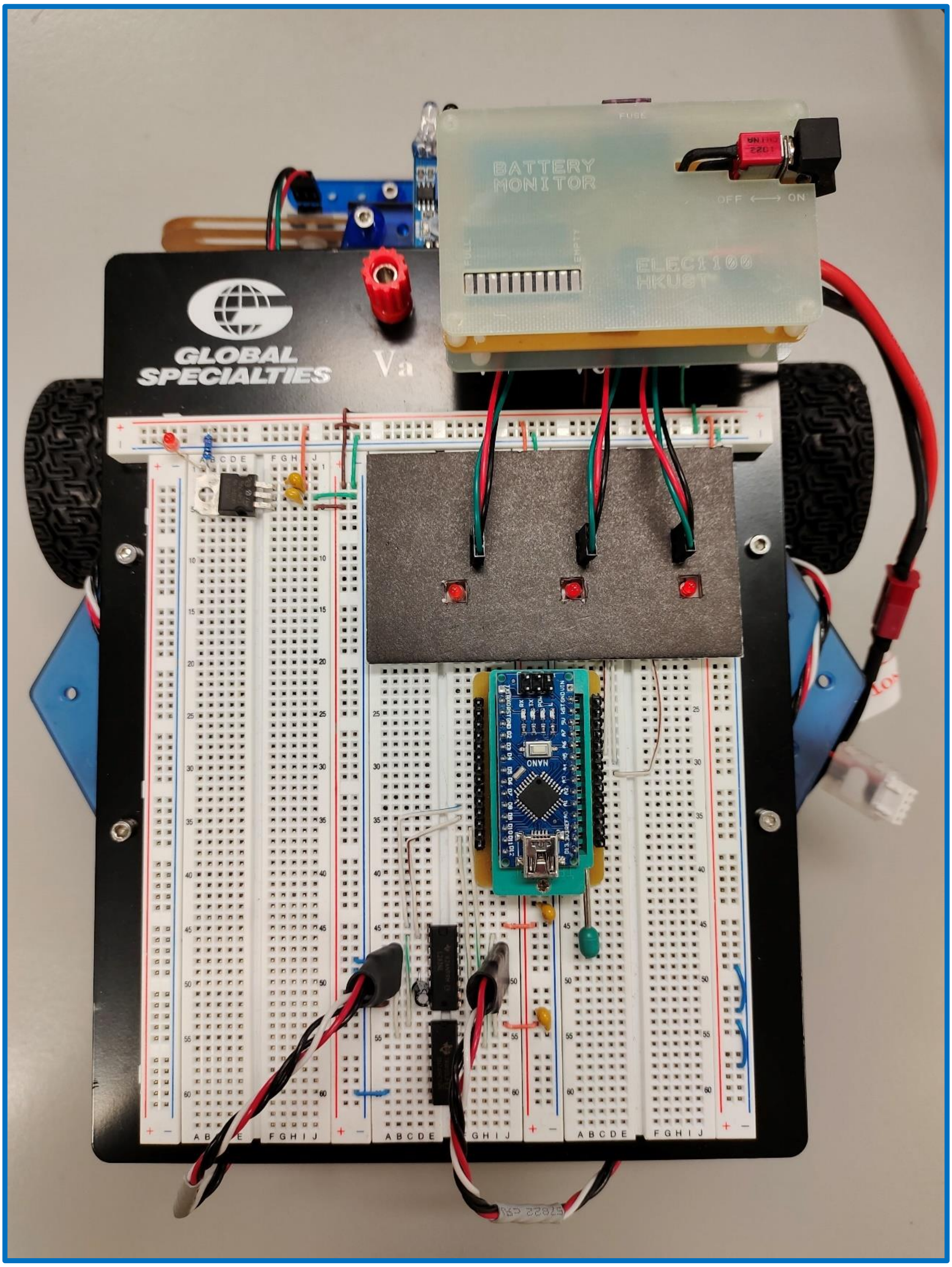
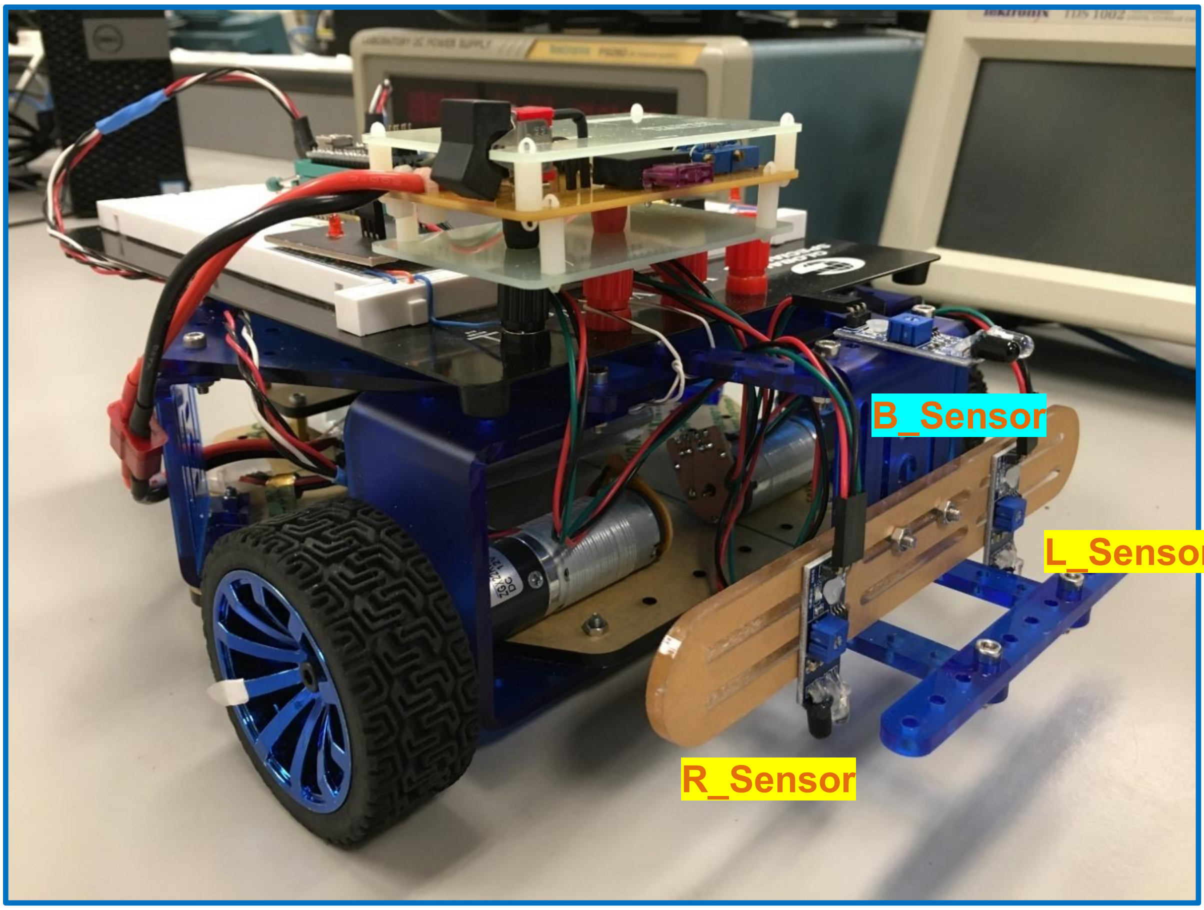


Photo 2: car sensors placement



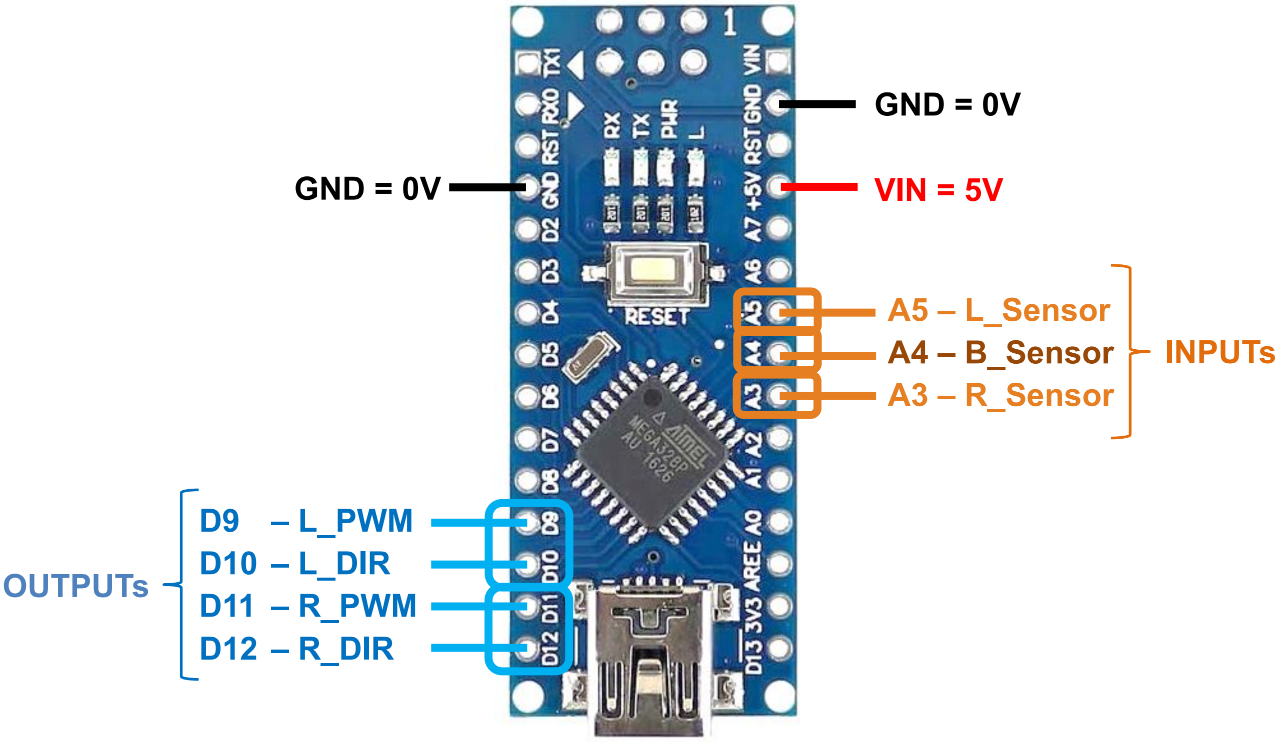
- About the **line sensors** (3 in total) on the car:
    - Each of them would give an output voltage when:
        - On white: output 0V
        - On black: output 5V
    - Line tracking sensors: for detecting the white lines on the demo mat.
        - **L_Sensor & R_Sensor** are connected to the Arduino input terminals **A5 & A3** as what you used in your lab#06 circuit.
    - Bumper sensor: for initiating the robot car at the "Start" line and for triggering the car at the "white wall" to run backward.
        - **B_Sensor** is connected to the Arduino input terminal **A4** as what you used in your lab#06 circuit.

- About the **Arduino board** on the car:
    - The Arduino board will be accepting signals from the line sensors **L_Sensor**, **B_Sensor** and **R_Sensor**.
    - The Arduino board will be controlling the motor's rotating directions **L_DIR** and **R_DIR**.
    - The Arduino board will be controlling the motor's speeds **L_PWM** and **R_PWM**.

Notes:

1. Arduino Uno-board in your Tinkercad simulation circuit is fairly similar to Arduino Nano-board. Nano-board (on the demo car) is preferred at the physical lab because of the small size.

2. Arduino Pins assignment is given as shown below and you **MUST** follow the given connection to write your control logic code.



Arduino Pins assignment in your Lab#06 & project circuit

- About your **Arduino** code:
  You will need to continually work on your Lab#06 code to write in more control statements for fulfilling the requirements in **Part E**.
  - Your code should read the sensor's signals **L_Sensor**, **B_Sensor** and **R_Sensor** as the logic inputs.
  - Your code should send logic outputs through **L_DIR** and **R_DIR** to control the rotating directions of motors.
  - Your code should send logic outputs through **L_PWM** and **R_PWM** to control the speeds of motors.

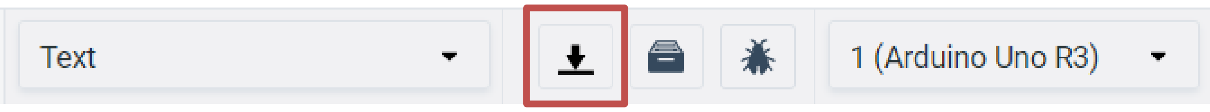
D) Code Preparation and Submission

Fill in your name before submitting your code to Canvas. This part of the code will be displayed on the screen at test & demo sessions, so you would know it's your demo trials on the camera.

```
/*
  ELEC1100 Your Lab#06 & Project Template

  To program the car tracking the white line on a dark mat

  Your name:

*/
```

You can still use Tinkercad coding "Text" section to do the writing and download your completed Arduino code from Tinkercad as shown below.

Download your completed Arduino code from Tinkercad



However, after downloading the code from Tinkercad, you should verify it using Arduino Software (IDE) such that it can be uploaded to the Nano Board without errors during the demo. The software can be downloaded from https://www.arduino.cc/en/software

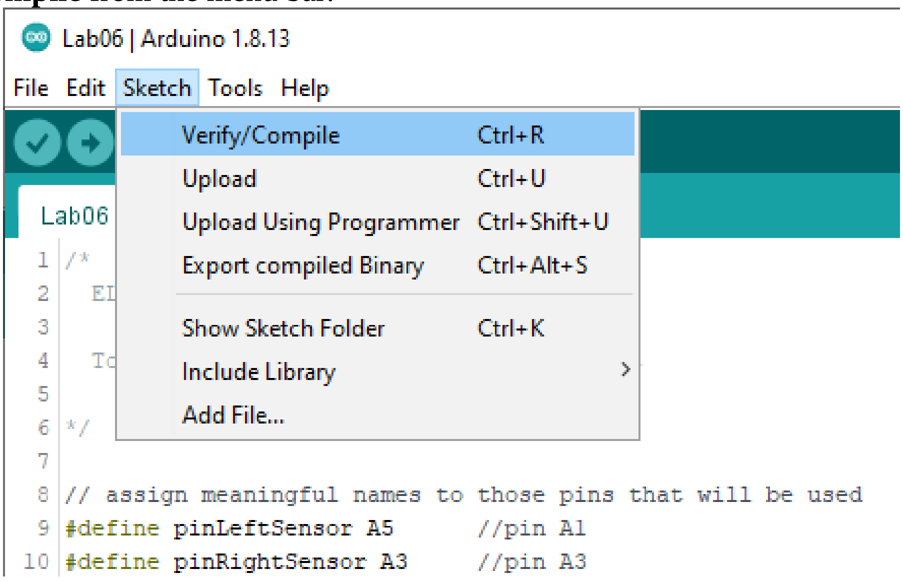
Arduino Software (IDE) on your computer desktop

Moreover, there is a constraint for the filename processed by Arduino Software. To avoid errors when opened by the software and for easy identification, you should rename your filename to this format:

<student id>_v<#>.

For example, if your student id is **12345678**, the submitted filename should be **12345678_v1**. If you submit two code versions, the second file name should be **12345678_v2**.

After the file is opened by the Arduino Software successfully, choose **Sketch → Verify/Compile** from the menu bar.



If there is no error, you should see something similar to this:

```
Done compiling.
Sketch uses 1160 bytes (3%) of program storage space. Maximum is 30720 bytes.
Global variables use 13 bytes (0%) of dynamic memory, leaving 2035 bytes for local variables. M
```

After done verification in Arduino Software, submit your code to your Canvas lab page (LA1/LA2/LA3) before the deadline of your own test & demo session.

- If you have two code versions, remember to click on both files when uploading. This is to let Canvas display both of your submissions on the same page.

LA1 (deadline 13:30pm), LA2 (deadline 09:00am), LA3 (deadline 12:00noon)

Late submission will NOT be accepted.

During the test & demo sessions (through ZOOM meeting), your TA will download your Arduino code from Canvas, upload it to the robot car, and start the trials on the demo mat at the physical lab for grading your coding work. This will be recorded in ZOOM meeting.

[Warning!!!]

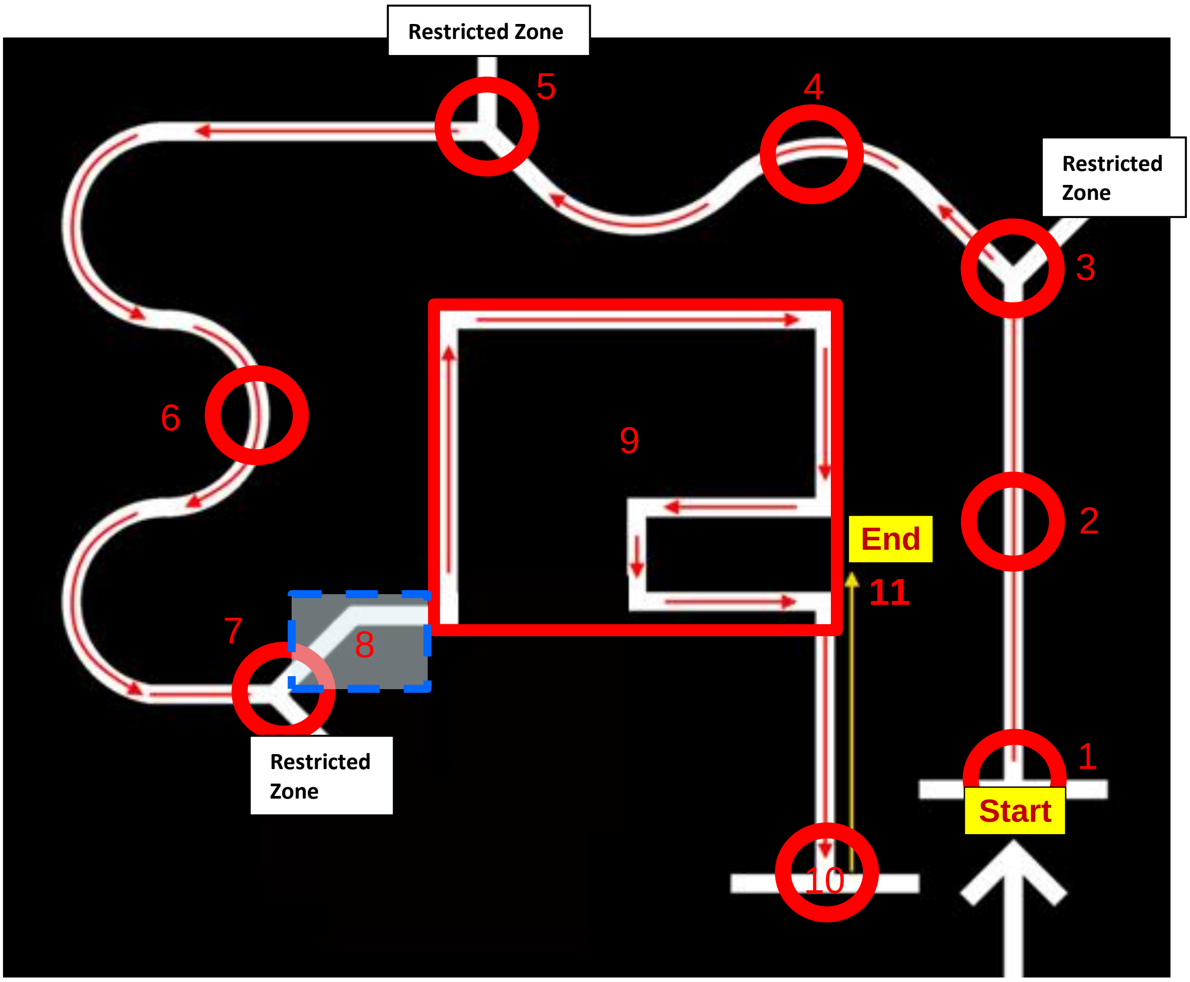
All your code submissions will be used to conduct a plagiarism check after the final demo. Copying from the others may result in a mark penalty or even failing this course.

E) Grading Policy for the Track Demo

With a correct Arduino code (your logic control unit), the robot car should complete the following tasks within each trial to achieve "**perfect run**":

- ✓ After power up, **the car wheels start turning but would stop in a few seconds**. So the robot car can be put at the START position such that both of the **left & right sensors** are **ON** the horizontal "Start" white line (*See Appendix: Car Specification*).
- ✓ By using a white paper to trigger the **bumper sensor**, the car should start to run.
- ✓ Once the car starts running, it tracks the white lines, and navigates through all of the splits and turnings.
- ✓ When the car successfully navigates to the left, after detecting the 3rd split, it should let itself stop within the blue square section (as shown below) for at least 1 second there, then resume moving and tracking the white lines.
- ✓ When the car arrives at the white wall, the **bumper sensor** shall be triggered again, and the car go backward until it reaches the "End" corner and stop there. Ensure the **bar of the left & right sensors** pass the horizontal line of the corner (*See Appendix: Car Specification*).

Points awarding scheme is as below.



Task No.	Task	Points
1	After power on, car wheels should be **STOP** at the beginning and **START** running after the bumper sensor is triggered by a white paper	**1**
2	Follow the straight Line	**1**
3	Navigate the 1st left split	**2**
4	Gentle curve I	**1**
5	Navigate the 2nd left split	**2**
6	Gentle curve II	**1**
7	Navigate the 3rd left split	**2**
8	Stop within the **blue square section** for at least 1 second and resumes to move	**2**
9	90-degree angle curves (finish all 7 angles)	**2**
10	Backward after the bumper sensor senses the white wall	**2**
11	Stop at the **End** corner (the bar of sensors must pass the horizontal line of this corner)	**2**
Time	Done all above as a "**perfect run**" within **25 seconds**	**2**
	Total	**20**

To join the **Coding Test 1**, each of you can submit at most 2 code versions.

	Coding Test 1	Submission Deadline	Trials
LA3	Apr 07 (Thu) 12:00-14:50	Apr 07 (Thu) **12:00**	**4** (2 trials each version)
LA2	Apr 12 (Tue) 09:00-11:50	Apr 12 (Tue) **09:00**	**4** (2 trials each version)
LA1	Apr 12 (Tue) 13:30-16:20	Apr 12 (Tue) **13:30**	**4** (2 trials each version)

To join the **Coding Test 2**, still, each of you can submit at most 2 code versions.

	Coding Test 2	Submission Deadline	Trials
LA2	Apr 19 (Tue) 09:00-11:50	Apr 19 (Tue) **09:00**	**4** (2 trials each version)
LA1	Apr 19 (Tue) 13:30-16:20	Apr 19 (Tue) **13:30**	**4** (2 trials each version)
LA3	Apr 21 (Thu) 12:00-14:50	Apr 21 (Thu) **12:00**	**4** (2 trials each version)

At your **Coding Test 1 & 2** sessions:

- If your code could already let the car achieve a "**perfect run**", you will receive a mark of 20 points (within 25 seconds) or 18 points (exceeds 25 seconds) as your demo score.
  - If you achieved 20 points, the full mark of project demo, you do not need to attend the following demo sessions.
- If not, your test results will not be graded, and you may come back later at the demo sessions with an improved coding work.

To join the **Early Demo**, still, each of you can submit at most 2 code versions.

	Early Demo	Submission Deadline	Trials
LA2	Apr 26 (Tue) 09:00-11:50	Apr 26 (Tue) **09:00**	**4** (2 trials each version)
LA1	Apr 26 (Tue) 13:30-16:20	Apr 26 (Tue) **13:30**	**4** (2 trials each version)
LA3	Apr 28 (Thu) 12:00-14:50	Apr 28 (Thu) **12:00**	**4** (2 trials each version)

To join the **Final Demo**, still, each of you can submit at most 2 code versions.

	Final Demo	Submission Deadline	Trials
LA2	May 03 (Tue) 09:00-11:50	May 03 (Tue) **09:00**	**4** (2 trials each version)
LA1	May 03 (Tue) 13:30-16:20	May 03 (Tue) **13:30**	**4** (2 trials each version)
LA3	May 05 (Thu) 12:00-14:50	May 05 (Thu) **12:00**	**4** (2 trials each version)

At your **Early & Final Demo** sessions:

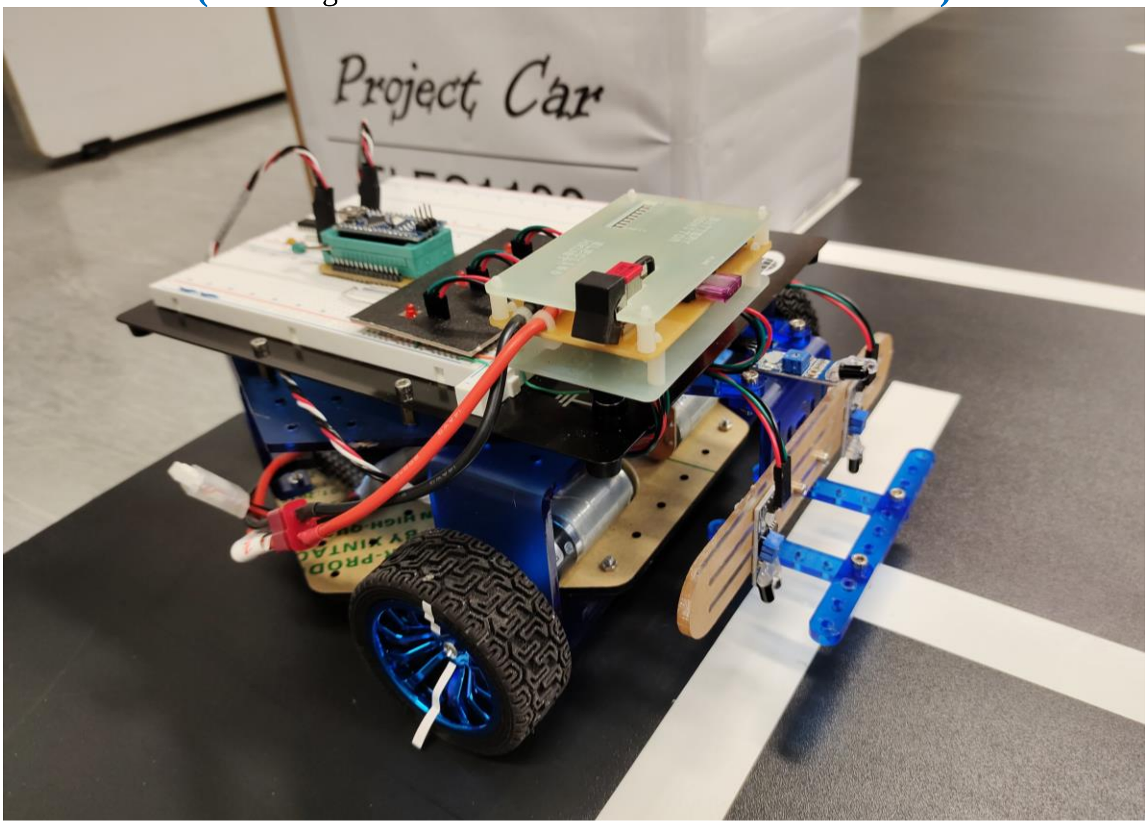
- If your code can let the car achieve a "**perfect run**", you will receive a mark of 20 points (within 25 seconds) or 18 points (exceeds 25 seconds) as your demo score.
- Otherwise, your score will be given depending on the **"Points awarding scheme"** from the "Start" white line within each trial in the order of achieving tasks No. 1-11.
- Fail at one task anywhere on the demo mat (touching the restricted zones, going out of the track, non-stop within the blue square section, etc.) will be regarded as **failing this trial**, and you will need to start from the very beginning as **the next trial**.
- **Your project demo score will be the highest one out of all your trials.**

F) Appendix: Car Specification

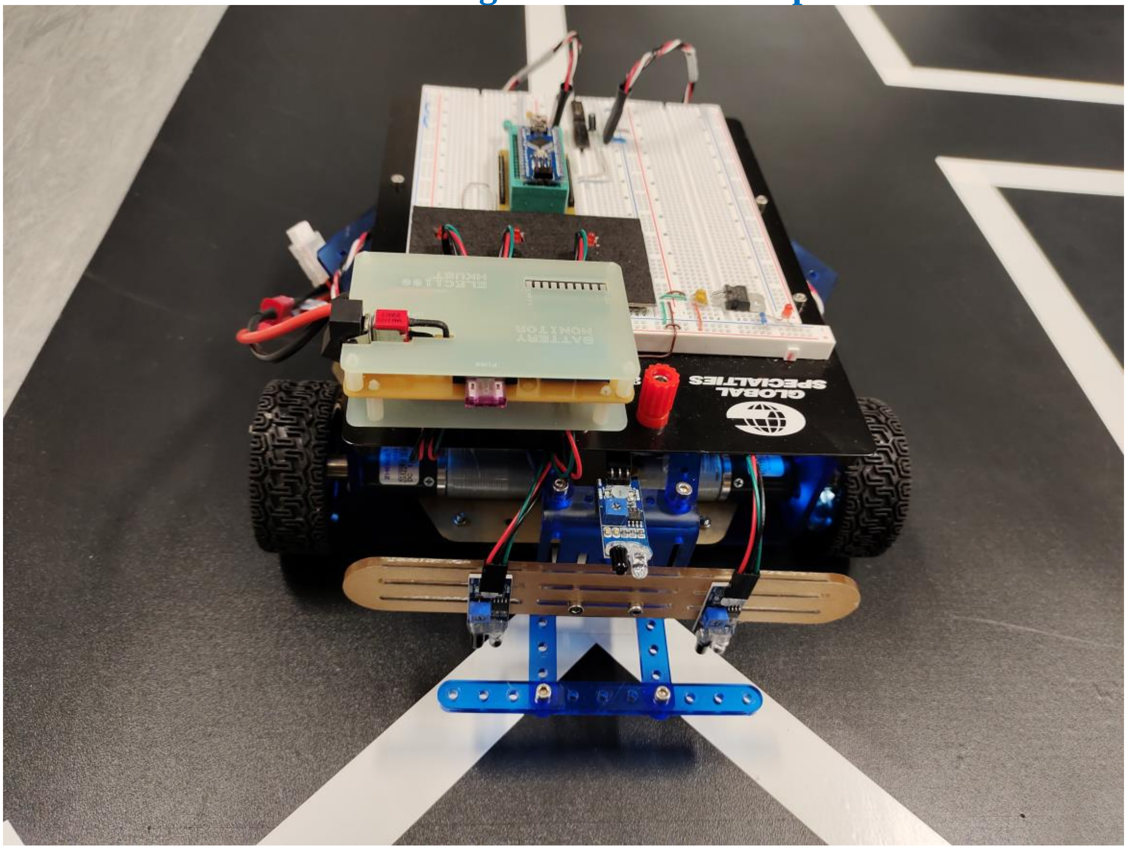
Below is for your reference to do the coding work.

Car at the "Start" white line

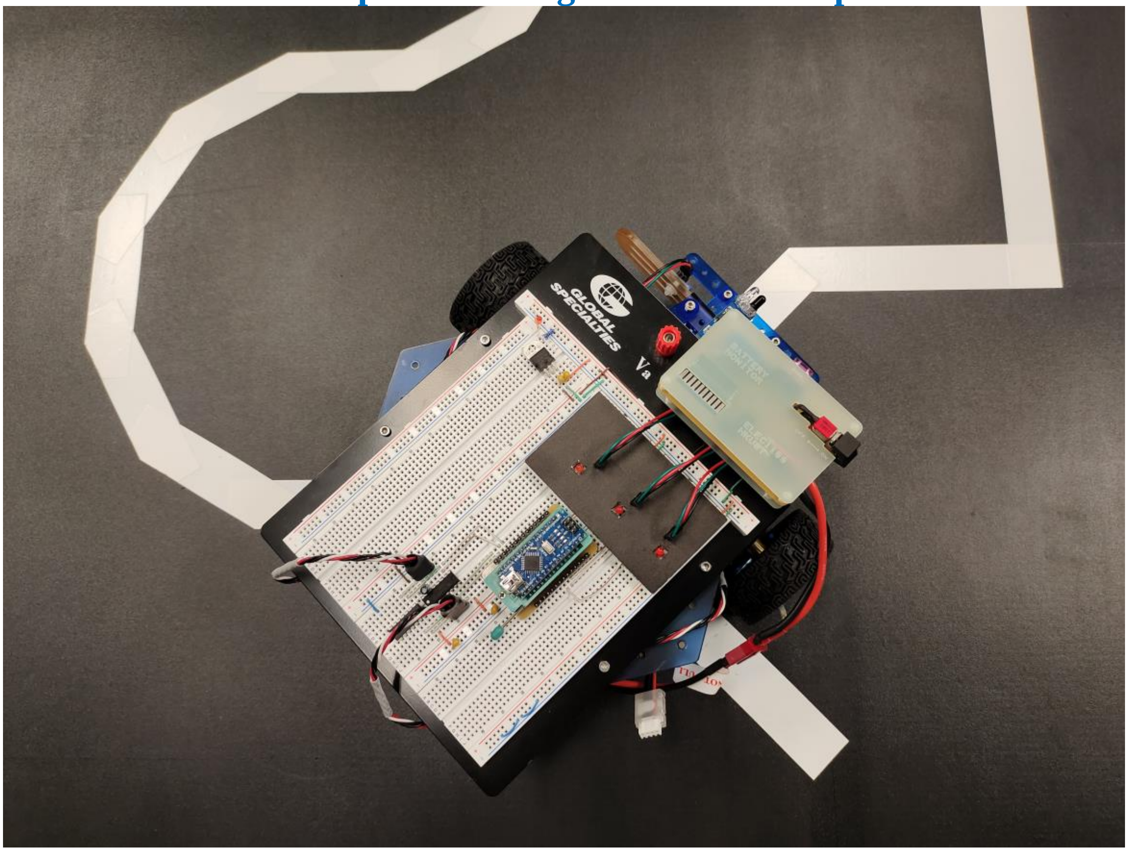
(Left & right sensors must above the horizontal white line)



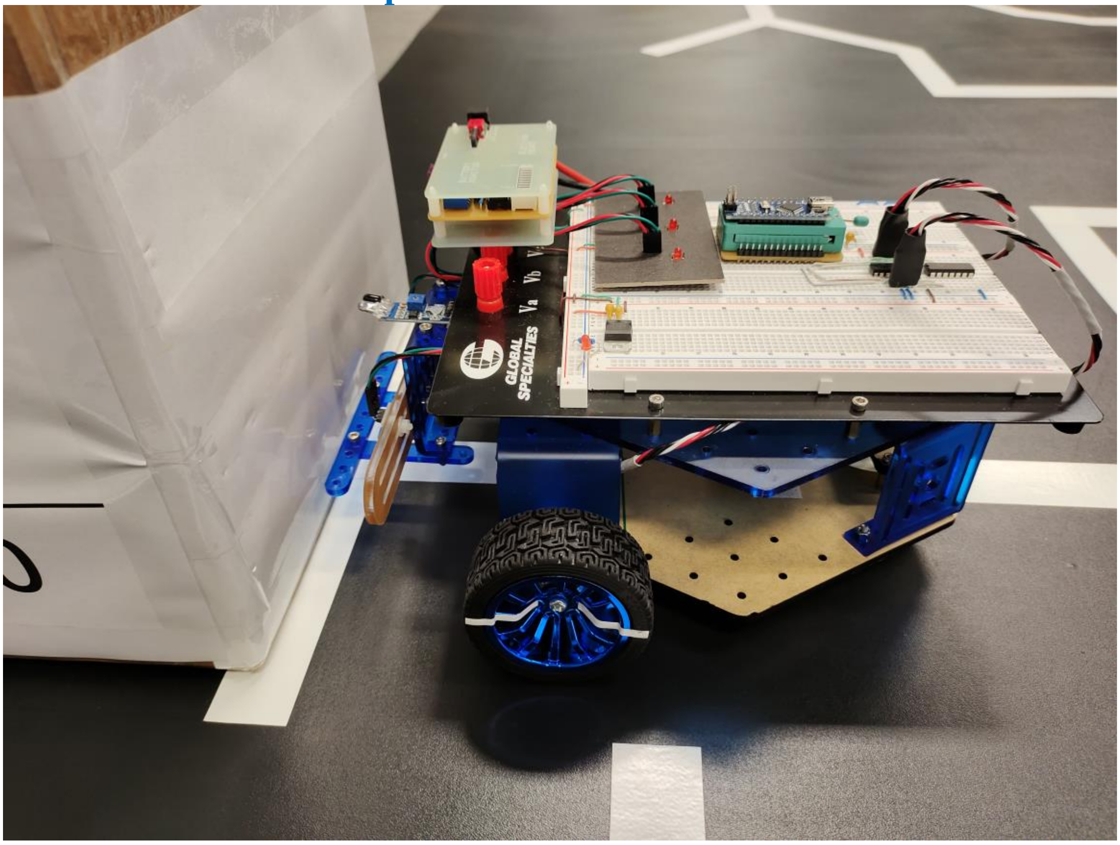
Line tracking sensors at the 1st split



Car stops after navigates the 3rd left split

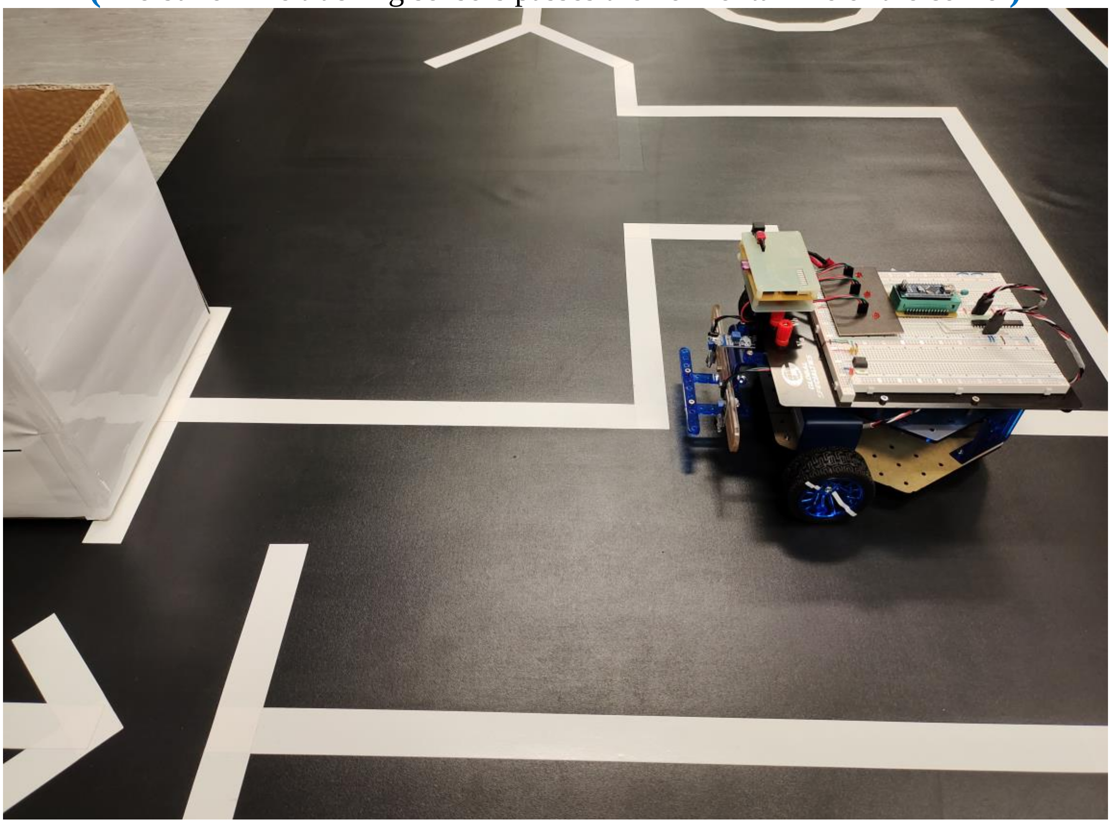


Bumper sensor at the white wall



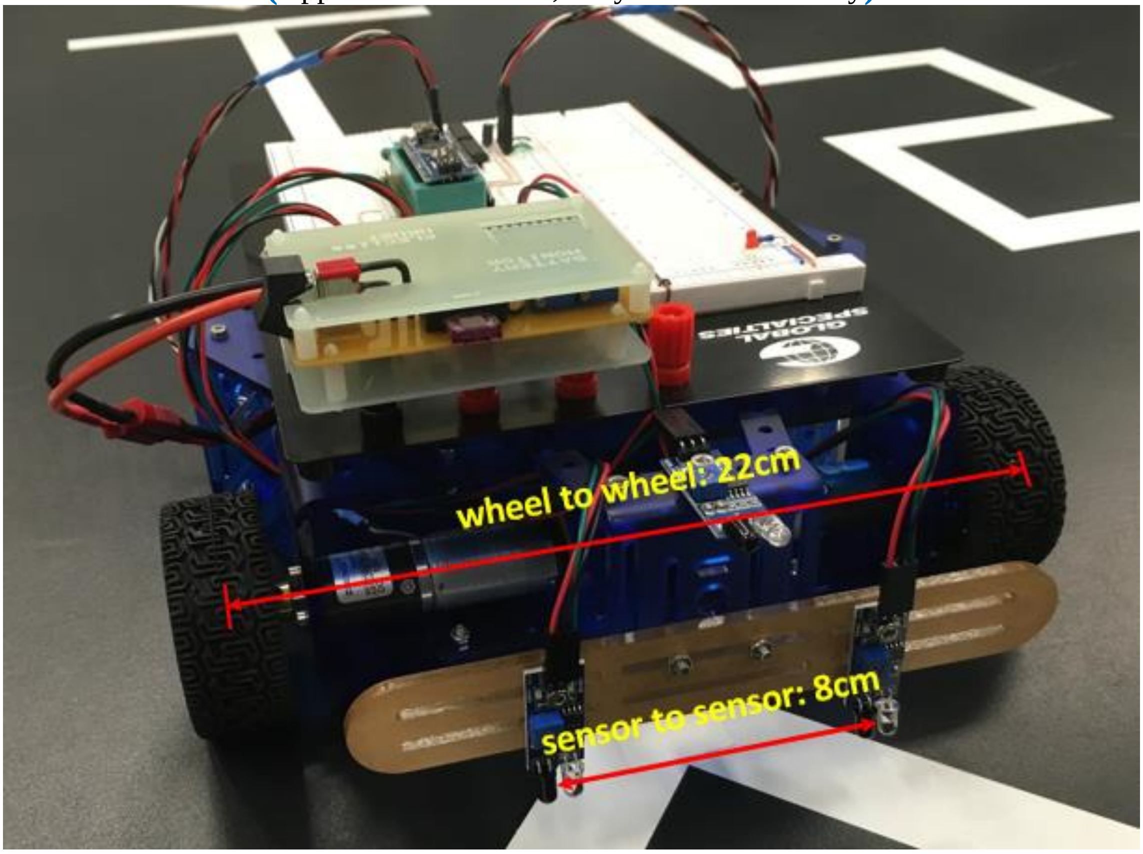
Car stops at the "End" corner

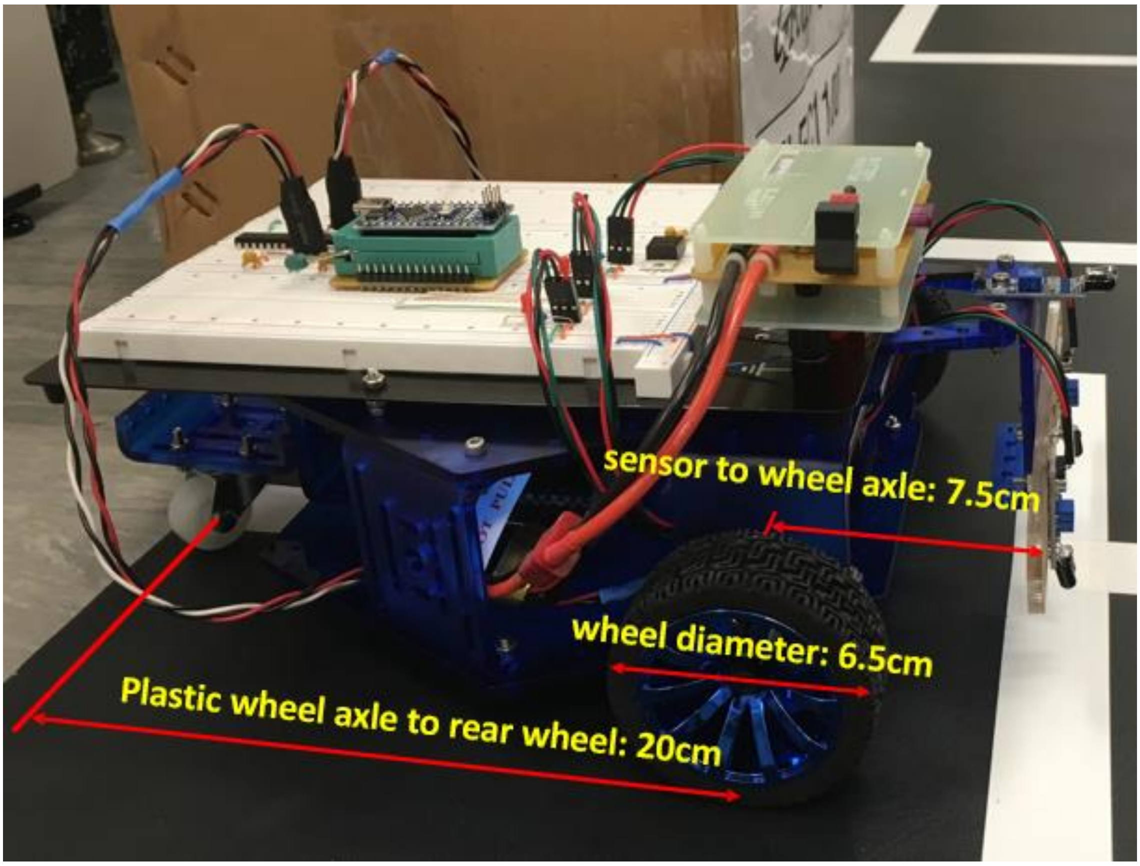
(The bar of line tracking sensors passes the horizontal line of the corner)



Car Dimension

(Approximated values, for your reference only)





Car Motor Speed

(Approximated values, for your reference only)

- PWM = 100% (255), ~60cm/s
- PWM = 50% (127), ~40cm/s

Notes:

1) The speed is only an average value and it may not be in linear relationship to duty cycle, most likely due to motor's inertia and discharge effect.
2) Due to the weight of the car, minimum PWM value shall be 100 to fulfill minimum motor starting power.

Your Project Demo Mat Scales