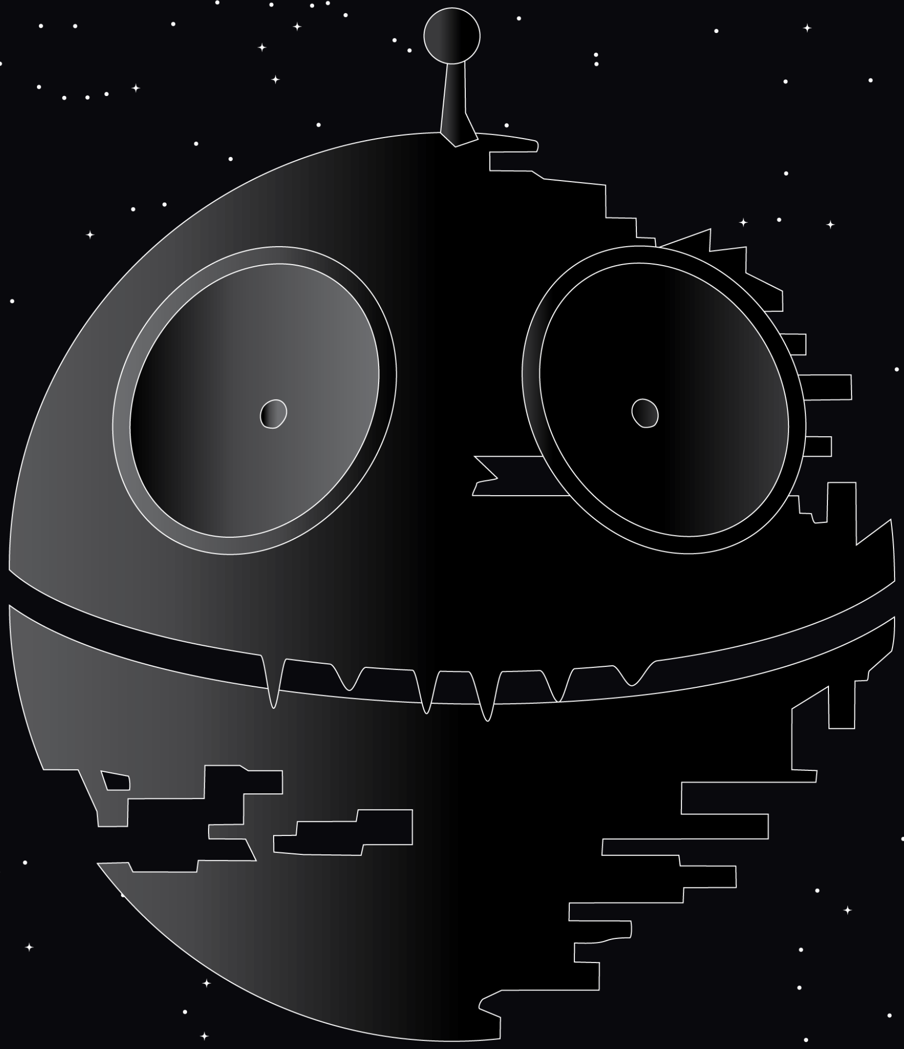
PRATT WARS

INTEGRATED DESIGN CHALLENGE

SP
20
26



ECE 110L: Integrated Design Challenge
Spring 2026

Contents

Pratt Wars

multicol

1 Introduction

Right now at a school not very far, far away...

It is a period of midterms. After countless hours in Perkins, student's studies have uncovered secret plans for the Administration's ultimate weapon, the Pratt Star, an armored robot student capable of ruining the curve for the entire school. One student smuggled these plans out of the library and into your hands. It is now up to you to put an end to this sinister plot.

In this high-stakes competition, your team of bots will come together to destroy the Pratt Star. Together, they will measure its weaknesses, develop a plan, and quell this sinister plot by striking where the Pratt Star is weakest. The curve of every engineering class rests in your hands, do you have what it takes? Will you be the heroes that the students in this department need?

1.1 The Pi Wings

Each team will pilot their own Pi Wing, a specialized bot equipped with sensors to detect various characteristics of the Pratt Star. Working as a team, your section will measure the magnetism of its armor, its ability to dissipate heat, its camouflage, its size, and its RF shielding. Using these values, your team will determine the Pratt Star's greatest weakness and strike there to take it down.

It will be up to you to construct and control your own Pi Wing. To accomplish your goal, your bot must be able to navigate complex tracks, detect objects, and communicate with the other bots on your team. No one bot can do this alone.

1.2 Learning Objectives

The Pratt Wars saga is more than just an exhilarating competition: It is a hands-on opportunity to develop foundational skills in engineering design, programming, and problem solving. Throughout the course of the challenge, you will:

- **Become comfortable using the Arduino IDE**
  You will gain fluency in the Arduino programming environment by writing, uploading, and debugging code that controls your bot's sensors, actuators, and behavior.
- **Develop thoughtful hardware design skills and make strategic choices**
  You will learn to evaluate, select, and integrate various hardware components, from motors to sensors, to build a reliable and efficient autonomous system. Critical thinking in hardware layout and modular design will be emphasized.
- **Understand how to troubleshoot effectively and efficiently**
  Problem solving is core to engineering. You will be challenged to diagnose issues quickly, apply systematic debugging techniques, and iterate on your design to ensure peak bot performance throughout the Grand Prix.

By the end of this experience, you will not only have contributed to a high-speed robotics showdown, but you will also have strengthened your toolkit as an engineer ready to tackle real-world challenges.

1.3 Overview of the IDC

Game Play

For each run:

- Each bot starts along its assigned path and checks five hash marks (1–5). These first five hashes determine the weakness of the Pratt Star's defenses, a number from 1–5, based on where the object is detected.
- At each hash mark, bots will
  - complete their sensing task and record the data;
  - The recorded data is the hash number where the object is present.
  - use their onboard RGB LEDs to signal their arrival at each hash mark:
    - Purple: 1st hash mark;
    - Green: 2nd hash mark;
    - Blue: 3rd hash mark;
    - Red: 4th hash mark;
    - White: 5th hash mark;
- After completing hash 5, each bot accelerates onto the final (trench-run) path and stops at its assigned hash.
- Each bot then transmits one value to the team:
  - If it detects an object at the final hash → transmit the measured value (1–5)
  - If no object is detected → transmit 0

  Think of it as: *Transmitted Value = (Object detected at trench-run hash)* × *(Weakness Value)*
- Bots flash their external RGB LEDs to indicate the presence of an object and communication states:
  - Green: object is detected at a hash mark;
  - Red: no object detected at a hash mark;
  - Blue: sending data to other bots;
  - White: receiving data from other bots
- Additionally, each bot will display the entire team's data in an array on an LCD after it has been received (checked at the conclusion of each trial run);
- Bots will determine the correct final task. This will be based on the sum of their team's values.

Final Task:

- Total damage score = sum of all 5 bots' transmitted values.
- If Total damage score is greater than or equal to 10 → the Pratt Star explodes and the team performs a celebratory Light Show
- If Total damage is less than or equal to 9 → the Pratt Star survives the attack and the bots perform a thematically appropriate song.

Oral Defense Prep For the first four weeks, students will complete individual discussion posts analyzing code logic and design strategies. These posts serve as training for an oral defense after the mid-point trial run, where students will respond to questions about system architecture, debugging processes, and optimization strategies.

Progress Video By Week 5, each team will submit a concise video (5–7 minutes) detailing key parts of their codebase, such as sensor polling routines, decision-making logic, and bot coordination. This video highlights how each team transformed individual code modules into a high-functioning sci-fi experience.

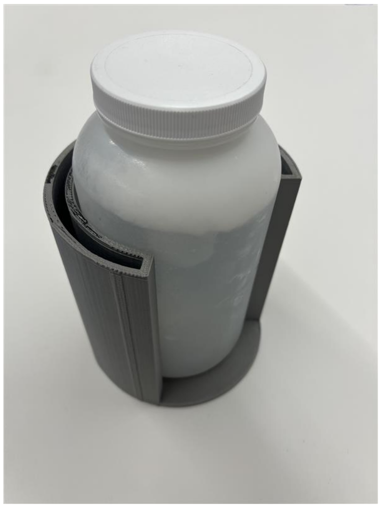
Oral Defense After the final demo, students will participate in a one-on-one five-minute long discussion. Their TA will individually ask them questions which they will answer. The questions will be based on the oral defense prep questions and similar to those shared in the question bank.

1.4 Scoring Values

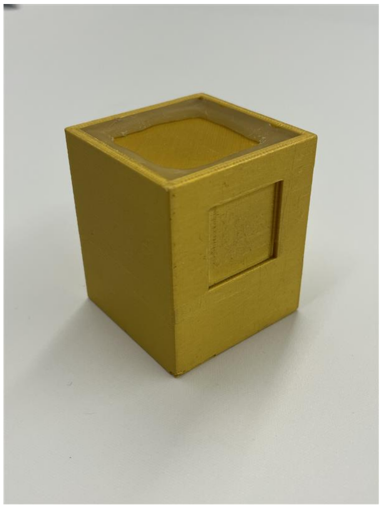
Score	Description
0	No Object at Final Hash Mark
1	Object placed at Hash Mark 1
2	Object placed at Hash Mark 2
3	Object placed at Hash Mark 3
4	Object placed at Hash Mark 4
5	Object placed at Hash Mark 5

These hash marks indicate each Pi Wing's measurement of the Pratt Star's weakness. Once a bot has its value measured, it will proceed to the trench run and attempt to attack the Pratt Star. Once each bot knows whether they hit their mark, they will communicate to the other Pi Wings how much damage they did to the Pratt Star.

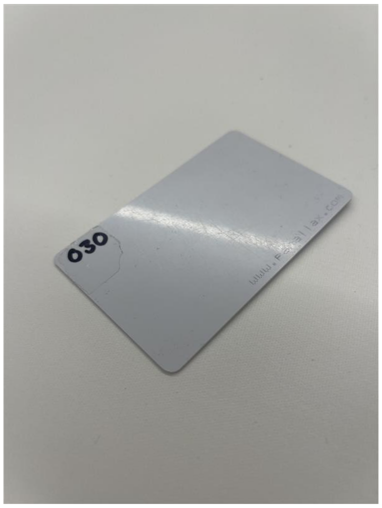
1.5 Objects and Game Board



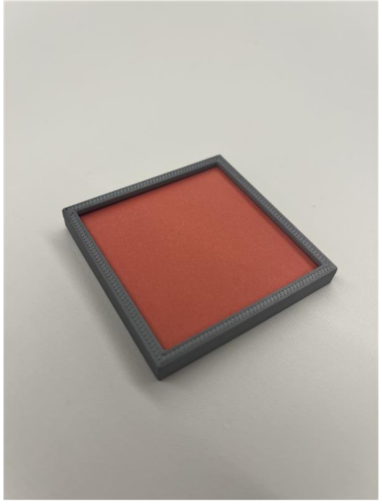
IR Therm Sensor: Heat Dissipation



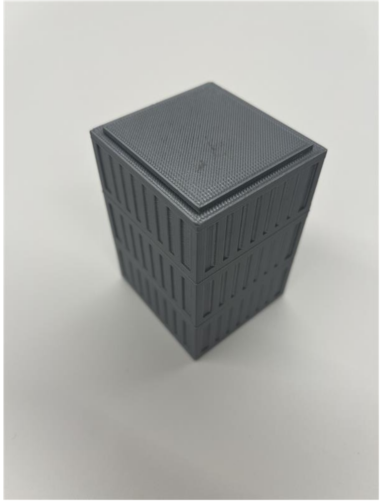
Hall Effect Sensor: Armor Magnetism



RFID Sensor: RF Shielding



Color Sensor: Camouflage



Ping Sensor: Armor Gap Size

Each image above shows an object that a bot must detect at the hash marks, triggering specific behaviors according to their sensor's logic and the scoring rubric.

2 Using This Manual: A Guide for Students

This guide is your team's official manual for success, designed to walk you through the entire experience from start to finish. Whether you're adjusting sensors, writing control algorithms, or preparing for the celebration, this manual will serve as your constant reference.

2.1 Weekly Lab Manuals

Each week of the challenge will include a dedicated lab manual, attached within this document as an internal section. These weekly lab manuals will:

- Break down the week's deliverables and how to complete them;
- Provide detailed technical and procedural guidance for your bot;
- Outline what is expected in the prelab, lab, and any post-lab activities;

Make sure to read your assigned lab manual carefully before each lab session. They are designed to not only help you complete your tasks efficiently, but also reinforce the core engineering concepts and skills you are expected to learn throughout the semester.

2.2 Appendix

At the end of this manual, you will find an appendix which consolidates key resources and reference material, including:

- A full semester timeline with important checkpoints and submission dates;
- Grading criteria and rubrics for each component of the challenge;
- Evaluation expectations for the Oral Defense and Progress Video;

Use the appendix as your primary reference for any questions about assessment, timelines, or expectations.

2.3 Reminder on Prompts

Throughout this document, there will be two additional prompts to let you know specific IDC work that needs to be completed. The two prompts are as follows, with N representing a number to help make sure you see all the prompts:

- **IDC Checkpoint (N):** Work to be shown to a TA during the IDC Checkpoint.
- **IDC Deliverable (N):** Other items to be completed and then included in the post-laboratory assignment.

2.4 Final Note

This is your playbook. Keep it close. Consult it often. And most of all, enjoy the journey as you build, plan, and innovate alongside your fellow Blue Devils.

3 IDC Week 1: Controlled Motion—Line Following

3.1 Abstract

As part of the IDC, your group will need to have your ’bot navigate paths marked with dark lines on a light background by using a QTI sensor. Along the paths, points of interest will be indicated by hash marks that are perpendicular to the line (and significantly wider than the line). Whenever you get to a hash mark, your ’bot will need to use its object sensor(s) to take a reading.

Note that this week, the work is divided into **four** sections - the Pre-Laboratory, the Experiment, the Exploration, and the IDC Checkpoint. IDC Checkpoints will specifically support the IDC Full System Demo. Although the first two IDC Checkpoints are intended to be accomplished by the groups individually, and are based on the success of a particular system or systems, the rest of the checkpoints will require the entire team to accomplish them.

3.2 Objectives

After performing this laboratory exercise, students should be able to build a ’bot that:

- (Exploration) Uses a set of three or four QTI (q=charge, t=transfer, i=infrared) sensors to navigate a track, and
- (IDC Checkpoint) Uses a set of three or four QTI sensors to navigate a track, stopping and performing tasks at hash marks.

3.3 Background

Take a look at https://pundit.pratt.duke.edu/wiki/Following_Lines for a general discussion of how to get a ’bot to follow a line.

3.4 QTI Sensor

Information about the specific QTI Sensor you are using is on the EGRWiki page at https://pundit.pratt.duke.edu/wiki/ECE_110/Equipment/QTI.

3.5 Pre-Laboratory Exercises

1. Read through the Pratt Pundit page on Following Lines.
2. Make sure you understand the concepts involved.
   - For this lab’s IDC Checkpoint, and later during the full system demo, the ’bot needs to follow a path with both straight sections and curves, and identify hash marks along that line.
   - The ’bot will gather information from three (or four) sensors then use that information to decide how (or if) to move.
   - When the ’bot encounters a hash mark it will stop and change the on-board RGB LED to a different color.

     Note: *Hash marks are more than an inch away from each other.*
3. Fill out a table that maps each of the eight (for three sensors) or sixteen (for four sensors) QTI states to some kind of action plan for the robot. Below are partially completed tables (3.5.2) for your reference.

   Note: *The tracks you are using will have both straight and curved sections, but no sharp corners (90° angles). The hash marks can show up anywhere on the track.*

3.5.1 Thinking Ahead

During the lab, you will be required to follow different types of paths:

1. (Exploration) Straight, circular, and s-curved lines.
   - For the straight and curved line, your 'bot needs to follow to the end; once it goes past the end of these lines, it can stop (i.e. if you have LLL or LLLL, the 'bot should stop).
   - For the circle, your 'bot needs to circumnavigate the path twice without stopping.
2. (IDC Checkpoint) Navigation–line following with hash marks on IDC specific path.
   - Your 'bot needs to stop at each hash mark for at least a quarter of a second before continuing down the path. When the 'bot gets to a hash mark, it should:

     (a) change the color of the on-board RGB LED:
     - Hash 1: Red
     - Hash 2: Green
     - Hash 3: Blue
     - Hash 4: Purple
     - Hash 5: White

     (b) turn off the on-board RGB LED before leaving, or after a quarter of a second, and

     (c) it should completely stop at the last hash mark.

3.5.2 QTI State Tables

	Left	Middle	Right	No.	Description	Action
3 QTIs	D	D	D	0	At a hash mark	stop, change light, wait, move straight past
	D	D	L	1		
	D	L	D	2	Highly unlikely	N/R
	D	L	L	3		
	L	D	D	4		
	L	D	L	5		
	L	L	D	6		
	L	L	L	7	Not on line	turn left and right until line is found

Pre-lab Deliverable (1): Create a list or table for the 3-sensor states and actions. This does not need to be in LaTeX but does need to clearly show the status of the three sensors, a description of what that status means, and a statement of the action your 'bot should take. If a state is "Highly unlikely," you can put that as a description with no action plan.

Pre-lab Deliverable (2): On Gradescope, answer the multiple choice questions about which actions your bot should perform when detecting certain patterns while using 4 QTI sensors.

Pre-lab Deliverable (3): On Gradescope, answer the multiple choice questions to fill in the blanks on the paragraph about QTI sensors and their functionality.

3.6 Pre-Laboratory Assignment

Your assignment responses should be uploaded to the ECE 110L Laboratory Gradescope site by the assignment deadline.

EACH STUDENT must submit their own INDIVIDUAL assignment responses!

You will respond to the following:

1. Type the Duke Honor Code:
   - *"I have adhered to the Duke Community standard in completing this assignment."*
2. **Pre-lab Deliverable 1:** Upload a list table for the 3-sensor states and actions.
3. **Pre-Lab Deliverable 2 and 3:** Answer the multiple choice questions on Gradescope.

Your assignment responses should be uploaded to the ECE 110L Laboratory Gradescope site by the assignment deadline.

EACH STUDENT must submit their own INDIVIDUAL assignment responses!

3.7 Equipment

- (1) CX-Bot
- Up to (4) QTI sensors with mounting hardware
- Up to (4) Black-Red-White cables
- (1) Parallax screwdriver

3.8 Exploration

For this exploration, you will use the QTI sensors to aid your 'bot in navigating a track.

1. Decide if you are going to use three or four QTI sensors for line following. There are advantages and disadvantages to each approach.
2. Mount the QTI sensors and connect them with the black-red-white cables to the dedicated BRW headers on the front of the CX-Shield.
   - Carefully note which pin numbers are on each red pin; the black pins are all connected to ground and the white pins are all connected to 5 V.
   - All three (or four) sensors should be right next to each other, and they should be collectively centered on the front of the 'bot. Leaving too much space between them is not a good idea.
3. Once they are mounted, copy the sample code found on the QTI Sensor Pratt Wiki page to a sketch and upload it to your 'bot. Link to Pratt Wiki in (3.4).
   - The sample code will look at one QTI and assumes it is connected to the BRW where red is 47. This should be your far-left QTI.
   - See what happens if you place the 'bot so that the QTI is over a light surface or a dark surface on a test mat.

   Checkpoint (1): Show your TA that you are correctly reading and displaying the information for one QTI.
4. Copy the sketch to a new sketch and change the code so it reads and displays information from all of your QTIs.
   - See what happens if you place the 'bot so that the QTIs are over different combinations of light surfaces or a dark surfaces on the mat.

   Checkpoint (2): Show your TA that you are correctly reading and displaying the information for multiple QTIs.

Discussion (1): Discuss the numerical values for the QTI when it is over a light surface versus a dark surface. Come up with what you think is a good threshold value between "dark" and "light."

5. Write code that converts your QTI states into a single number.

   Arduino can do math with digital logic - anything true is considered a 1 if you do math with it, and anything false is considered a 0.

   - Assume each QTI's dark/light response can be converted to a binary 0 or 1 (if the decay time is less than or above the threshold you established above).
   - Next assume the far left QTI's state is the most significant bit and the far right QTI's state is the least significant bit.
   - With that, you can convert the three or four binary "digits" into a single decimal value as follows:

     (a) For three QTIs:

     ```
             int state = 4*sLeft + 2*sMiddle + sRight;
     ```

     where `sLeft`, `sMiddle`, `sRight` are logical expressions about whether the left, middle, or right sensor is reading light or not. For example, you could replace `sLeft` with `qti1 > 2000`.[1]

     (b) For four QTIs:

     ```
             int state = 8*sFLeft + 4*sMLeft + 2*sMRight + sFRight;
     ```

     where `sFLeft`, `sMLeft`, `sMRight`, `sMRight` are logical expressions about whether the far left, middle left, middle right, or far right sensor is reading light or not.

   Checkpoint (3): Show your TA that your code is correctly reading the QTIs and converting them to single overall state value for several combinations of dark and light.

6. Write code that uses a `switch...case` structure.

   Now that your 'bot can figure out what state it is in, it is time for it to do something about it!

   - There is a structure in Arduino code known as a `switch...case` statement that will be very useful here. When you have a variable that contains an integer that can be mapped to a particular action item, a `switch...case` structure is a much more efficient way to go than a large if tree.

     Hint:*Take a look at https://www.arduino.cc/reference/tr/language/structure/control-structure/switchcase/ for the syntax and examples.*

[1] That would be a particularly poor choice of threshhold, but you get the idea.

- If you have multiple cases that are all meant to do the same thing, you can stack several case statements before the code each is meant to run:

```
switch (var) {
    case 0:
    case 3:
        // stuff for either case 0 or 3
        break;
    case 1:
    case 2:
    case 6:
        // stuff for case 1, 2, or 6
        break;
    default:
        // stuff for any other value of var
        break;
    }
```

- Here is where the work on the pre-lab comes into play! Figure out what you want your robot to do for those states you think you might achieve. For the default case, you should have your robot stop moving and you should print a message to the serial monitor essentially indicating that the 'bot does not know what to do there. Every other case should be specifically addressed in the `switch` structure.

- Write some functions you can call that will have the robot move forward, turn or veer left, turn or veer right, and stop.

  Hint:*You performed all these tasks in a previous lab. It's time to modify them!*

  Note:*For this exploration, if your 'bot identifies a hash mark, it should keep going straight. For IDC Checkpoint 1 that will no longer be the case.*

7. Once your code is written, test it on the Straight line track, Circle track, and (optional) squiggle track.

   - **DO NOT** stop on the hash marks and **DO NOT** write code to find a line for the LLL or LLLL cases.
     - The LLL or LLLL case should be coded to cause the robot to just stop.
   - There are two main items you will need to tweak in your code once you get the `switch` sorted out:
     - How much time does a 'bot spend moving in a particular way before you check the state again?
     - How fast should each wheel be going for each maneuver? ***Hint:*** *Full speed for long periods of time between checks is not a good idea...*

   Checkpoint (4): Show your TA that your 'bot can navigate circle path. To get full credit, it must complete two full revolutions, one clockwise and one counterclockwise. (the TA will manually reverse the 'bot's direction).

   Deliverable (1): Having completed the previous checkpoint, you can respond "Yes" to the yes or no question on Gradescope for this deliverable. **Responding "Yes" and not having your checkpoint confirmed by your TA *may* result in a missed point, so make sure your TA has confirmed the previous checkpoint!**

   Note: *The IDC Checkpoint code will very much be based on this code; you will want to document this simpler version first. Have two files, not just one!*

3.9 IDC Checkpoint

As noted in the Pre-Laboratory exercise and in the IDC documentation, IDC Checkpoint 1 is all about your 'bot's navigation (AKA Line Following). You already have the code to follow a line while ignoring hash marks, now consider which overall QTI state value or values mean you have found a hash mark and should thus stop the 'bot and change the color.

- When your bot reaches a hash mark, it should stop.
  - Hash 1: Purple
  - Hash 2: Green
  - Hash 3: Blue
  - Hash 4: Red
  - Hash 5: White
- The bot should pause at hash marks 1-4 long enough to identify the correct RGB color before moving on. The RGB LED should remain on between 0.25 seconds and 1.0 second before turning off again.
- After the RGB LED flashes at the last hash mark it should remain off and the 'bot should not move again.

Finally, the bot must move off of its 5th hash and on to the "trench run". This is the final path, with a single hash for each bot. Importantly, bots are not required to stop at any hash on the trench run except for the one corresponding to their sensor. No penalty will be given for stopping at all hashes, but it is not required.

IDC Checkpoint (1): Show your TA that your 'bot can start moving at the beginning of your group's IDC path, stop at the hash marks for the appropriate amount of time, turn on the on-board RGB LED the appropriate color for the appropriate amount of time, and then turn off the on-board RGB LED before moving forward again. After the 5th hash, your bot must leave its original path and navigate to its final location on the trench run.

IDC Deliverable (1): Include your code for having the CX-Bot follow the line, change the color of the on-board RGB LED accordingly. **Be sure your code is thoroughly commented.** This *does not mean every line needs a comment*, but there should be enough comments to make it clear what your code does. Don't forget about the standards we use and found in the Arduino Coding Guide.

3.10 IDC Oral Defense Prep

To prepare you for the IDC Oral Defense, you will have multiple questions at the end of each IDC Week Lab. Thorough and clear answers to the questions below are worth a single point. Answers that are not on topic, thorough, generally lacking effort, or not answering a question will result in zero points.

1. How, generally, do QTI sensors distinguish between light and dark surfaces, and how does this concept connect to what the acronym QTI stands for?
2. What are some assumptions that you made when deciding how to mount QTIs on your bot?
3. Describe your understanding of the rcTime function.

IDC Deliverable (2): In a single paragraph, answer the above questions on Gradescope.

3.11 Assignment

Your assignment responses should be uploaded to the ECE 110L Laboratory Gradescope site by the assignment deadline.

EACH STUDENT must submit their own INDIVIDUAL assignment responses!

You will respond to the following:

Type the Duke Honor Code:

- *”I have adhered to the Duke Community standard in completing this assignment.”*

Deliverable 1: Simple Line Following 3.8

- Respond “Yes” to the question “Did your TA mark off your group for finishing checkpoint 4?” in the provided box on Gradescope
- Be sure your TA has confirmed you Checkpoint 4 completion.

IDC Deliverable 1: Code for Line Following with hash marks 7.7

- Upload your code for following lines, stopping at hash marks, turning on the appropriate RGB LED colors, and turning them off before continuing. Code should show that the ’bot does not move past the final hash mark. Make sure your TA has marked your group as finishing IDC Checkpoint 1.
- Be sure your code is thoroughly commented. This does not mean every line needs a comment, but there should be enough comments to make it clear what your code does. Don’t forget about the standards we use and found in the Arduino Coding Guide.

IDC Deliverable 2: IDC Oral Defense Prep 6.9
On Gradescope, answer the following questions in a single paragraph:

- How, generally, do QTI sensors distinguish between light and dark surfaces, and how does this concept connect to what the acronym QTI stands for?
- What are some assumptions that you made when deciding how to mount QTIs on your bot?
- Describe your understanding of the rcTime function.

4 IDC Week 2: Sensing and Transmitting

4.1 Abstract

Now that you have used the full set of sensors and gotten the basics of line following to work separately, it is time to put those together! The main task for this week will be to integrate line following with your group's sensing task. You will also need to wirelessly send a value indicating the information obtained about your target object to a sentry 'bot, which the TAs will provide. In a future lab, you and your team will implement ways to communicate this information to all of the bots in your lab section.

- Work this week is concentrated in one part - the IDC Checkpoint - which will specifically support the integrated design challenge and will be graded based on your 'bot's ability to complete the task using a particular system or set of systems.
- This week's IDC Checkpoint is intended to be accomplished by the groups individually and will require your group to properly implement their own unique systems involving sensors and the communication device.
  - If your group finishes before lab is over, you are expected to work with other teams to help them implement their systems and plans without doing the work for them.
- IDC Checkpoints 3 and 4 will require collaboration with, and completion as, a team as you will design and apply general communication systems.
- This lab has brief pre- and post-laboratory deliverables, but does not contain any deliverables for the experiments or explorations.

Note: *Completion of IDC Checkpoint 1 is crucial before attempting IDC Checkpoint 2.*

4.2 Objectives

After performing this laboratory exercise, students should be able to build a 'bot that:

- Navigates the IDC track(s) they have been assigned
- Stops at hash mark(s) along the track
- Uses the appropriate color of the on-board RGB LED to indicate the hash mark
- Uses the appropriate color of an external RGB LED to indicate sensing and transmitting
- Uses the sensor they have been assigned to locate a target object and reject other objects, and
- Communicates the target object's information.

4.3 Pre-Laboratory Exercises

Reference Lab 1 and Lab 3, as needed. Be comfortable with both.

1. Talk with your partner(s) about:
   - how IDC Week 1 lab went for you and your section. If tasks were not completed, plan for any tasks that still need to be resolved.
   - a way for using your sensors during this lab, and
   - XBee communication and how your section/team can use single characters to transmit information about each bot and the outcome of what is being sensed.

2. Once you feel confident in your group's ability to tackle system integration, write down any notes from your communication with your lab partner(s) and compile them into a paragraph.

Pre-lab Deliverable (4): Write a brief paragraph detailing any ideas or notes that came up in your communication with your partners.

Pre-lab Deliverable (5): During IDC Week 1, your group was assigned a sensor to work with. Describe in a few sentences how you should connect this sensor to your Bot and how this sensor works (What it is doing to be able to measure what it is supposed to measure).

Pre-lab Deliverable (6): On Gradescope, answer the multiple choice questions about structuring your IDC code.

Pre-lab Deliverable (7): On Gradescope, answer the multiple choice questions about the ASCII representation of characters.

4.4 Pre-Laboratory Assignment

Your assignment responses should be uploaded to the ECE 110L Laboratory Gradescope site by the assignment deadline.

EACH STUDENT must submit their own INDIVIDUAL assignment responses!

You will respond to the following:

1. Type the Duke Honor Code:
   - *"I have adhered to the Duke Community standard in completing this assignment."*
2. **Pre-lab Deliverable 1:** Write a brief paragraph describing your conversation with your partner(s) and any important notes you may have.
3. **Pre-lab Deliverable 2:** Describe how you should connect this sensor to your Bot and how this sensor works.
4. **Pre-lab Deliverable 3 and 4:** Answer the multiple choice questions on Gradescope.

Your assignment responses should be uploaded to the ECE 110L Laboratory Gradescope site by the assignment deadline.

EACH STUDENT must submit their own INDIVIDUAL assignment responses!

4.5 Experimental Exercise

1. As a section/team talk about communication.
2. Share your response to the Pre-Lab Assignment as it pertains to any of the ideas you and your partner(s) had about communication.
   - Remember that five 'bots will be, potentially simultaneously, sending information about their target object.
   - Focus your discussion around figuring out how each Xbee can understand what is going on given that it is entirely possible two or more 'bots will be "speaking" at once.

4.6 Exploration

1. Continue building a 'bot that accomplishes your group's navigation, sensing, and communication tasks.
2. Use an external RGB LED (not the on-board) in your 'bot's mini-breadboard to indicate the sensing and transmission processes.
   - Transmitting/Sending Data: Blue, flashing during communication code
   - Receiving Data: White, flashing during communication code (used next week)
   - +Sense (object detected/correct object): Green, 0.25 second flash
   - -Sense (object missing/incorrect object): Red, 0.25 second flash

3. Test your ’bot on the actual track, but make sure you are charging your battery pack so you have plenty of stored energy for the IDC Checkpoint.
    - Use wall power when testing to keep your batteries fresh for the checkpoint, which will require your ’bot to use the batteries only.

Note: *The IDC document is a useful resource as you develop your code or adjust hardware.*

4.7 IDC Checkpoint

For IDC Checkpoint 1, you learned about QTIs and how to use them to follow a path. You also learned how to make your on-board RGB LED turn on and off specific colors as it reached different hash marks.

Note: *It is crucial for your group to have completed IDC Checkpoint 1 before attempting IDC Checkpoint 2 since you will be integrating code needed for both.*

1. Integrate your code for navigation and hash mark indication with sensing.
2. Integrate this new code with data transmission code.

IDC Checkpoint (2): Show your TA that your ’bot can navigate as it did for IDC Checkpoint 1 while also using it’s sensor to look for the object your group is expected to identify. Once your ’bot has finished navigating and sensing it must also use an XBee module to transmit (send) the correct data (see IDC document). The sentry ’bot will be used to identify the correct values are being transmitted/sent.

IDC Deliverable (3): This is the integrated code for navigating your group’s path, sensing objects, and transmitting the appropriate data value for your IDC task (location, count, etc). **Be sure your code is thoroughly commented.** This *does not mean every line needs a comment*, but there should be enough comments to make it clear what your code does. Don’t forget about the standards we use and found in the Arduino Coding Guide.

4.8 IDC Oral Defense Prep

To prepare you for the IDC Oral Defense, you will be asked multiple questions at the end of each IDC Lab. Thorough and clear answers to the questions are worth a single point each. Answers that are not on topic, thorough, generally lacking effort, or which fail to answer a question will receive zero points.

1. How does your group's sensor generally function?
2. What are some of the design decisions your group made when mounting your sensor and what variables impacted these decisions?
3. What kind of data does your group's sensor send to the Arduino and how does the Arduino (in code) process this data?

IDC Deliverable (4): In a single paragraph, answer the above questions on Gradescope.

4.9 Assignment

Your assignment responses should be uploaded to the ECE 110L Laboratory Gradescope site by the assignment deadline.

EACH STUDENT must submit their own INDIVIDUAL assignment responses!

You will respond to the following:

1. Type the Duke Honor Code:
   - *"I have adhered to the Duke Community standard in completing this assignment."*
2. **IDC Deliverable 2: Code for Navigation, Sensing, and Transmitting** 7.7
   - Upload the integrated code for navigating your group's path, sensing objects, and transmitting the appropriate data value for your IDC task.
   - Be sure your code is thoroughly commented. This does not mean every line needs a comment, but there should be enough comments to make it clear what your code does. Don't forget about the standards we use and are found in the Arduino Coding Guide.
3. **IDC Deliverable 2: IDC Oral Defense Prep** 6.9
   On Gradescope, answer the following questions in a single paragraph:
   - How does your group's sensor generally function?
   - What are some of the design decisions your group made when mounting your sensor and what variables impacted these decisions?
   - What kind of data does your group's sensor send to the Arduino and how does the Arduino (in code) process this data?

5 IDC Week 3: Receiving and Displaying

5.1 Abstract

Now that your ’bot can successfully complete its sensing task and transmit data, it is time to get it to display data it receives. The main task for this week will be to integrate code for receiving data from other ’bots and displaying all necessary data/information (see IDC document) on a LCD.

- Work this week is once again focused on completed an IDC Checkpoint designed to specifically support the integrated design challenge and will be graded based on your ‘bot’s ability to complete the task using a particular system or set of systems.
- This week’s IDC Checkpoint will require your group to properly implement the team’s agreed upon system for communicating and displaying data on a LCD.
- This checkpoint (IDC Checkpoint 3), and the next checkpoint (IDC Checkpoint 4), require collaboration with, and completion as a team.
- This lab has brief pre- and post-laboratory deliverables, but does not contain any deliverables for the experiments or explorations.

Note: *Completion of IDC Checkpoint 1 and 2 is crucial before attempting IDC Checkpoint 3.*

5.2 Objectives

After performing this laboratory exercise, students should be able to build a ’bot that:

- Navigates the IDC track they have been assigned,
- Stops at hash mark(s) along the track while flashing the on-board RGB LED the correct colors
- Uses the sensor they have been assigned to locate a target object and reject other objects while using flashing an external RGB LED the correct colors to confirm sensing
- Transmit a target object’s information to all other ‘bots AND receive target object information from other ‘bot’s.
- Display all necessary information on a LCD.

5.3 Pre-Laboratory Exercises

1. Talk with your partner(s) about:
    - how IDC Week 2 lab went for you and your section. If tasks were not completed, plan for any tasks that still need to be resolved.
    - a way for improving Xbee communication, and
    - how your to create an array to display on your LCD.
2. Once you feel confident in your group’s ability to tackle system integration, write down any notes from your communication with your lab partner(s) and compile them into a paragraph.

Pre-lab Deliverable (8): Write a brief paragraph detailing any ideas or notes that came up in your communication with your partners.

Pre-lab Deliverable (9): On Gradescope, answer the multiple choice questions about topics relevant to the IDC.

5.4 Pre-Laboratory Assignment

Your assignment responses should be uploaded to the ECE 110L Laboratory Gradescope site by the assignment deadline.

EACH STUDENT must submit their own INDIVIDUAL assignment responses!

You will respond to the following:

1. Type the Duke Honor Code:
   - *"I have adhered to the Duke Community standard in completing this assignment."*
2. **Pre-lab Deliverable 1:** Write a brief paragraph describing your conversation with your partner(s) and any important notes you may have.
3. **Pre-lab Deliverable 2:** Answer the multiple choice questions on Gradescope.

Your assignment responses should be uploaded to the ECE 110L Laboratory Gradescope site by the assignment deadline.

EACH STUDENT must submit their own INDIVIDUAL assignment responses!

5.5 Experimental Exercise

1. As a section/team continue talking about communication
2. Share your response to the Pre-Lab Assignment as it pertains to any of the ideas you and your partner(s) had about adding on to the communication code to allow it to also receive.
3. Share your response to the Pre-Lab Assignment as it pertains to any of the ideas you and your partner(s) had about displaying the appropriate data on the LCD at the appropriate time and flashing the external RGB LED appropriately.

5.6 Exploration

1. Continue building a ‘bot that accomplishes your group’s navigation, sensing, communication (transmitting AND receiving), and displaying tasks.
2. Continue using the external RGB LED in your ‘bot’s mini-breadboard to indicate the sensing, transmitting, and NOW receiving processes.
   - Transmitting/Sending Data: Blue, flashing during communication code (last week)
   - Receiving Data: White, flashing during communication code
   - +Sense (object detected/correct object): Green, 0.25 second flash
   - -Sense (object missing/incorrect object): Red, 0.25 second flash
3. Confirm your LCD functions properly.
   - Backlight is turned off
   - Switches on back are set to 9600 baud rate
   - Red, black, and white cables are plugged in correctly AND to the dedicated shield pins
   - Use of nylon washers or plastic brackets when mounting, LCD is not shorting out
4. Test your ’bot on the actual track, but make sure you are charging your battery pack so you have plenty of stored energy for the IDC Checkpoint.
   - Use wall power when testing to keep your batteries fresh for the checkpoint, which will require your ’bot to use the batteries only.

Note: *The IDC document is a useful resource as you develop your code or adjust hardware.*

5.7 IDC Checkpoint

For IDC Checkpoint 2, you integrated sensing and transmitting with your navigation code, making your on-board RGB LED turn on and off specific colors as it reached different hash marks and your external RGB LED flash specific colors when your sensor detected objects an you transmitted data.

Note: *It is crucial for your group to have completed IDC Checkpoints 1 and 2 before attempting IDC Checkpoint 3 since you will be integrating code needed for all of three of them.*

1. Integrate your code for navigating, sensing, and transmitting with receiving code.

2. Integrate this new code with code for displaying the necessary information on your LCD.

IDC Checkpoint (3): Show your TA that your 'bot can navigate as it did for IDC Checkpoint 1 and 2 while also using it's sensor to look for the object your group is expected to identify. It must use an XBee module to transmit the correct data (see IDC document) similar to how it did in IDC Checkpoint 2, but this time you must also receive data from it and display the IDC appropriate data on your LCD (see IDC document).

IDC Deliverable (5): Integrate your code for sensing, navigating, and transmitting with your code for receiving and for displaying data on a LCD (see IDC document). Save this as a separate file. **Be sure your code is thoroughly commented.** This *does not mean every line needs a comment*, but there should be enough comments to make it clear what your code does. Don't forget about the standards we use and found in the Arduino Coding Guide.

5.8 IDC Oral Defense Prep

To prepare you for the IDC Oral Defense, you will be asked multiple questions at the end of each IDC Lab. Thorough and clear answers to the questions are worth a single point each. Answers that are not on topic, thorough, generally lacking effort, or which fail to answer a question will receive zero points.

1. How does the Xbee's serial buffer function? Knowing how it works, what are some considerations you and your section should make in terms of sending data?
2. Describe the system you are using to encode your bot's identity and sensing object information. What are some benefits and drawbacks to this system?
3. Describe how your code receives and stores data from other bots. What kinds of code structures are you using and why?

IDC Deliverable (6): In a single paragraph, answer the above questions on Gradescope.

5.9 Assignment

Your assignment responses should be uploaded to the ECE 110L Laboratory Gradescope site by the assignment deadline.

EACH STUDENT must submit their own INDIVIDUAL assignment responses!

You will respond to the following:

1. Type the Duke Honor Code:
   - *"I have adhered to the Duke Community standard in completing this assignment."*
2. **IDC Deliverable 1: Code for Navigation, Sensing, Transmitting and Receiving** 7.7
   - Upload the integrated code for navigation and sensing, transmitting to and receiving from all team 'bots, and displaying the required information on a LCD.
   - Be sure your code is thoroughly commented. This does not mean every line needs a comment, but there should be enough comments to make it clear what your code does. Don't forget about the standards we use and are found in the Arduino Coding Guide.
3. **IDC Deliverable 2: IDC Oral Defense Prep** 6.9
   On Gradescope, answer the following questions in a single paragraph:
   - How does the Xbee's serial buffer function? Knowing how it works, what are some considerations you and your section should make in terms of sending data?
   - Describe the system you are using to encode your bot's identity and sensing object information. What are some benefits and drawbacks to this system?
   - Describe how your code receives and stores data from other bots. What kinds of code structures are you using and why?

6 IDC Week 4: Data Analyzing and Calculating

6.1 Abstract

Now that your 'bot can successfully navigate, sense, transmit, receive and display all necessary data/information, it is time to get it to analyze this data and calculate how to perform its final routine. The main task for this week will be to integrate code for analyzing the data it collected and the data it received from other bots so it can determine its next maneuver. This varies based on the theme of the IDC, so be sure to reference this semesters IDC document.

- Work this week is focused on completing the final IDC Checkpoint designed to specifically support the integrated design challenge and will be graded based on your 'bot's ability to complete EVERY task using a particular system or set of systems.
- This week's IDC Checkpoint will require your group to properly implement the team's agreed upon system for analyzing and calculating the data transmitted and received.
- This checkpoint (IDC Checkpoint 4) requires collaboration with, and completion as as team.
- This lab has brief pre-and post-laboratory deliverables, but does not contain any deliverables for the experiments or explorations.

Note: *Completion of IDC Checkpoints 1–3 are crucial before attempting IDC Checkpoint 4.*

6.2 Objectives

After performing this laboratory exercise, students should be able to build a 'bot that:

- Navigates the IDC track they have been assigned.
- Stops at the hash mark(s) along the track while flashing the on-board RGB LED the correct colors.
- Uses the sensor they have been assigned to locate a target object and reject other objects while flashing an external RGB LED the correct colors to confirm sensing.
- Transmits a target object's information to all other 'bots AND receive target object information from other 'bots.
- Displays all necessary information on a LCD.
- Analyzes the data it collected and it received to determine the result.
- Calculates the final steps the 'bot must perform.

6.3 Pre-Laboratory Exercises

This lab is all about integrating the ability to finish the IDC event.

1. Talk with your partner(s) about:
   - the past 3 weeks of labs, and any issues pertaining to line following, sensing and detecting objects, XBee communication, and displaying with an LCD that still need solutions.
   - ideas you have for helping your 'bot analyze the data it collected and it received, and what to do with that data (***see IDC document for details)***. Be sure to post on Ed as needed.
2. Once you feel confident in your group's ability to tackle system integration, write down any notes from your communication with your lab partner(s) and compile them into a paragraph.

Pre-lab Deliverable (10): Write a brief paragraph detailing any ideas or notes that came up in your communication with your partners.

Pre-lab Deliverable (11): On Gradescope, answer the multiple choice questions about Arduino coding.

6.4 Pre-Laboratory Assignment

Your assignment responses should be uploaded to the ECE 110L Laboratory Gradescope site by the assignment deadline.

EACH STUDENT must submit their own INDIVIDUAL assignment responses!

You will respond to the following:

1. Type the Duke Honor Code:
   - *"I have adhered to the Duke Community standard in completing this assignment."*
2. **Pre-lab Deliverable 1:** Write a brief paragraph describing your conversation with your partner(s) and any important notes you may have.
3. **Pre-lab Deliverable 2:** Answer the multiple choice questions on Gradescope.

Your assignment responses should be uploaded to the ECE 110L Laboratory Gradescope site by the assignment deadline.

EACH STUDENT must submit their own INDIVIDUAL assignment responses!

6.5 Final Task

Once every bot has evaluated the location of its object and transmitted this information to the team, each bot should display an array containing the team's scores on an LCD screen. This display needs to be visible at the end of each trial (i.e., it will be checked after celebrations are performed) and remain on the LCD screen until students are told to reset their bots.

Now in possession of the team's full data set, the bots will analyze the data and determine the next steps. The bots will calculate the sum of the scores reported by all groups and calculate the average bot score using the following equations,

$$S = G_1 + G_2 + \cdots + G_n$$

where G_i is the score reported by group i and n is the total number of groups in the section. Once S is calculated, the outcome is determined by the following criteria:

- S is greater than 10 $\Rightarrow$ the Pratt Star explodes and the team performs a celebratory Light Show
- S Less Than or Equal to 9 $\Rightarrow$ the Pratt Star survives the attack and the bots perform a thematically appropriate song.

Light Show: Using your onboard RGB LED, external RGB LED, and any other LEDs you would like to attach to your bot, program a 5–10 second pre-scripted light display.

Song: Using the LCD (with piezoelectric speaker) mounted to your bot, your section should play 10–20 seconds of a pre-programmed tune.

6.6 Experimental Exercise

1. As a section/team talk about the final task(s) your ‘bots must perform.
2. Share your responses to the Pre-Lab Assignment as it pertains to any of the ideas you and your partner(s) had about analyzing and calculating data.
3. Reference the IDC document as there may be specific requirements, physical switches implemented, or other tasks not highlighted in the lab manual.

4. Be prepared to have your ’bots coordinate with each other.

6.7 Exploration

1. Continue building a ‘bot that accomplishes your group’s navigation, sensing, communication, displaying, and final tasks (AKA Endgame).
2. Implement any code/functions or physical items necessary to complete the endgame tasks.
3. Confirm that your ‘bot can complete all endgame tasks while not moving and continuing to display all data necessary on the LCD until asked to reset your ‘bot.
4. Test your ‘bot on the actual track, but make sure you are charging your battery pack so you have plenty of stored energy for the IDC Checkpoint.
    - Use wall power when testing to keep your batteries fresh for the checkpoint, which will require your ‘bot to use the batteries only.

Note: *The IDC document is a useful resource as your develop your code or adjust hardware.*

6.8 IDC Checkpoint

For IDC Checkpoint 3, you integrated receiving and displaying with your existing code, displaying the appropriate data and flashing both the on-board and external RGB LED the appropriate colors at the apporpriate times.

Note: *It is crucial for your group to have completed IDC Checkpoints 1–3 before attempting IDC Checkpoint 4 since you will be integrating code needed for all of four of them.*

1. Integrate your code for navigating, sensing, transmitting, receiving, and displaying with your code for analyzing the data and calculating what to do for the final tasks.
2. Integrate this new code with code for completing the IDC.

IDC Checkpoint (4): Show your TA that your ’bot can navigate, sense, communicate (and display) as it did for IDC Checkpoint 3, but this time your ’bot must be able to analyze the data and use that data to calculate what to do next. Essentially this is a full system integration. See the IDC document to determine exactly what that is.

IDC Deliverable (7): Integrate your code for sensing, navigating, transmitting, and receiving with your code for analyzing, calculating and completing the final tasks (see IDC document). This should be based on the analysis and calculation of the data transmitted/received (i.e. collected) by the team of ‘bots. Save this as a separate file. **Be sure your code is thoroughly commented.** This *does not mean every line needs a comment*, but there should be enough comments to make it clear what your code does. Don’t forget about the standards we use and are found in the Arduino Coding Guide.

6.9 IDC Oral Defense Prep

To prepare you for the IDC Oral Defense, you will have multiple questions at the end of each IDC Week Lab. Thorough and clear answers to the questions are worth a single point each. Answers that are not on topic, thorough, generally lacking effort, or not answering a question will result in zero points.

1. After you obtained sensor information from other bots, how did your bot determine which of the three celebrations to implement?
2. How did your bot determine at what point to begin the celebration?
3. How does your bot address not receiving a response from one or more bots in the section? If you do not implement this, what might you go about doing?

IDC Deliverable (8): In a single paragraph, answer the above questions on Gradescope.

6.10 Assignment

Your assignment responses should be uploaded to the ECE 110L Laboratory Gradescope site by the assignment deadline.

EACH STUDENT must submit their own INDIVIDUAL assignment responses!

You will respond to the following:

1. Type the Duke Honor Code:
   - *"I have adhered to the Duke Community standard in completing this assignment."*
2. **IDC Deliverable 1: Code for Navigation, Sensing, Transmitting, Receiving, Displaying, and Analyzing/-Calculating** 7.7
   - Upload your most current integrated code for completing IDC Checkpoint 4.
   - Be sure your code is thoroughly commented. This does not mean every line needs a comment, but there should be enough comments to make it clear what your code does. Don't forget about the standards we use and are found in the Arduino Coding Guide.
3. **IDC Deliverable 2: IDC Oral Defense Prep** 6.9
   On Gradescope, answer the following questions in a single paragraph:
   - After you obtained sensor information from other bots, how did your bot determine which of the three celebrations to implement?
   - How did your bot determine at what point to begin the celebration?
   - How does your bot address not receiving a response from one or more bots in the section? If you do not implement this, what might you go about doing?

7 IDC Week 5: Design Revising and Finalizing

7.1 Abstract

Now that your ’bot can successfully complete the IDC, it is time to make sure it can do so with consistency and style. The main task for this week will be to finish any remaining IDC Checkpoints, fine tune your integrated code, and integrate any code your group or team deems necessary to meet remaining IDC standards. This varies based on the theme of the IDC, so be sure to reference this semester’s IDC document.

- Work this week is ***NOT*** focused on completing a particular IDC Checkpoint, but rather it is designed to give your team time to revise their ’bot design and their code, while also finalizing any unfinished checkpoints.
- You must be able to demonstrate that your integrated code has the ability for your bot to complete the full system demo while not hindering the other ’bots.
- This lab has brief pre- and post-laboratory deliverables, but does not contain any deliverables for the experiments or explorations.

Note: *Completion of IDC Checkpoints 1-4 are crucial for completing IDC Week 6 Lab and the full system demo. Your team’s groups will help each other solve any remaining problems and with implementing creative features that set your team apart (see IDC document for any possible or specific requirements–creativity does not equate decorations or labels on ’bots).*

7.2 Objectives

After performing this laboratory exercise, students should have a ’bot that:

- Navigates the IDC track they have been assigned.
- Stops at the hash mark(s) along the track while flashing the on-board RGB LED the correct colors.
- Uses the sensor they have been assigned to locate a target object and reject other objects while flashing an external RGB LED the correct colors to confirm sensing.
- Transmits a target object’s information to all other ’bots, receives target object information from other ’bots, AND displays all necessary information on a LCD.
- Analyzes the data it collected and it received to determine the result before calculating the final steps the ’bot must perform.
- Implements any remaining tasks, refines creativity, and repeatedly demonstrates an ability to complete the final goals of the IDC (see IDC document).

7.3 Pre-Laboratory Exercises

This lab is all about putting final touches on your ’bot and testing to ensure that it can complete the full system demo consistently while meeting all requirements (see IDC document).

1. Talk with your partner(s) about:
    - the past 4 weeks of labs, and any issues pertaining to the completion of the full system demo.
    - you have for helping your ’bot complete the full system demo. Be sure to post on Ed as needed.
2. Once you feel confident in your group’s ability to act on a plan to complete the full system demo, write down any notes from your communication with your lab partner(s) and compile them into a paragraph.

Pre-lab Deliverable (12): Write a brief paragraph detailing any ideas or notes that came up in your communication with your partners.

Pre-lab Deliverable (13): On Gradescope, answer the multiple choice questions about topics relevant to the IDC Full System Demo.

7.4 Pre-Laboratory Assignment

Your assignment responses should be uploaded to the ECE 110L Laboratory Gradescope site by the assignment deadline.

EACH STUDENT must submit their own INDIVIDUAL assignment responses!

You will respond to the following:

1. Type the Duke Honor Code:
   - *"I have adhered to the Duke Community standard in completing this assignment."*
2. **Pre-lab Deliverable 1:** Write a brief paragraph describing your conversation with your partner(s) and any important notes you may have.
3. **Pre-lab Deliverable 2:** Answer the multiple choice questions on Gradescope.

Your assignment responses should be uploaded to the ECE 110L Laboratory Gradescope site by the assignment deadline.

EACH STUDENT must submit their own INDIVIDUAL assignment responses!

7.5 Experimental Exercise

1. As a section/team talk about the requirements to complete the full system demo, and continue developing a plan to complete with minimal errors.
2. Share your responses to the Pre-Lab Assignments it pertains to any of the ideas you and your partner(s) had about analyzing and calculating data.
3. Work with your teammates to find solutions to problems your group may still have with communication, displaying the necessary data, or other theme specific tasks (see IDC document). You should determine what each group should accomplish in Lab, and how you can help each other succeed.

Note: **Your IDC score depends on the success of your team, not just your 'bot. Now is the time to ensure that your 'bots can reliably complete the full system demo.**

7.6 Exploration

1. Fine tune and finalize your bot so it can accomplish any remaining IDC Checkpoints or required creative tasks, and is ready for the Full System Demo.
2. Practice repeatedly AS A TEAM!
3. Test your 'bot on the actual track, but make sure you are charging your battery pack so you have plenty of stored energy when practicing.

7.7 IDC Deliverables

This section is here just to remind you to finish any remaining checkpoints your group or team need to finish.

IDC Deliverable (9): Update and submit a copy of your IDC code as a .ino file. This code should be as fully integrated as possible and should contain the entirety of your teams remaining code/solutions. Save this as a separate file. **Be sure your code is thoroughly commented.** This *does not mean every line needs a comment*, but there should be enough comments to make it clear what your code does. Don't forget about the standards we use and that are found in the Arduino Coding Guide.

7.8 Assignment

Your assignment responses should be uploaded to the ECE 110L Laboratory Gradescope site by the assignment deadline.

UNLIKE the other labs this semester, you should USE THE GROUP SUBMISSION FEATURE on Gradescope to submit this week's assignment with your IDC group!

You will respond to the following:

1. Type the Duke Honor Code:
   - *"I have adhered to the Duke Community standard in completing this assignment."*
2. **IDC Deliverable 1: Improved Code for Full System Integration** 7.7
   - Upload your group's IMPROVED code for full system integration, which should include everything the 'bot must do to complete the entirety of the IDC's required tasks, including any changes from the previous iteration and the EXPECTED performances based on the analysis and calculations. *If the code is the same, comment this at the top of your code and then upload that version before you submit it.*
   - Be sure your code is thoroughly commented. This does not mean every line needs a comment, but there should be enough comments to make it clear what your code does. Don't forget about the standards we use and that are found in the Arduino Coding Guide.
3. **IDC Oral Defense Video**
   - Instead of writing a discussion post this week, **as a group** you will upload a 5-7 minute long video according to the detailed rubric on the IDC Week 5 Canvas page. After you record your video, you can post it **as a group submission** on Gradescope. One of you must submit it and add the others.

Your assignment responses should be uploaded to the ECE 110L Laboratory Gradescope site by the assignment deadline.

8 IDC Week 6: Full System Demo and Oral Defense

8.1 Abstract

Now that your bot has had the opportunity to finalize every task, it is time for the **"FULL SYSTEM DEMO"**.

- Work this week focuses on completing the five trials of your **"FULL SYSTEM DEMO"**. Ten minutes after your lab section starts, during which time your team should make last-minute preparations, you will be asked to charge your bots and direct your attention to the lab staff for instructions leading up to the start of the demo.
- In between trials...
  - you **will** be allowed to make physical adjustments to your 'bot, such as tightening a screw, adjusting the position of a sensor, or even fixing a wire.
  - you **will not** be allowed to upload any code. Once trial one has started, that code will be used through the entire **"FULL SYSTEM DEMO"**.
  - you are encouraged to charge your batteries.
- After the last trial of the **"FULL SYSTEM DEMO"**, students will meet their TA individually in the common area outside the lab for oral defense (see the IDC document).
- Students who have finished the oral defense, or are waiting to provide their oral defense, will work on removing all sensors, wires, and components in general that they added to their 'bot and CX-Shield.
- This lab has brief pre- and post-laboratory deliverables, but does not contain any deliverables for the experimental exercise.

8.2 Objectives

After performing this laboratory exercise, students should have a 'bot that:

- can complete the **"FULL SYSTEM DEMO"** (see the IDC document).

Students should also prepare themselves for the Oral Defense by:

- having a deep understanding of the code used to program their 'bot,
- knowing why each decision was made, how their sensor was used, and which components they added, and
- working with their partner(s) to determine any final preparation needed.

8.3 Pre-Laboratory Exercises

This lab is all about completing the **"FULL SYSTEM DEMO"**.

1. Talk wiht your partner(s) about:
   - the **"FULL SYSTEM DEMO"** requirements,
   - how your code works,
   - and what last-minute adjustments to the code or sensors need to be made for the demo.
2. Once you feel confident in your group's ability to act on a plan, write down any notes from you communication with your lab partner(s) and compile the into a paragraph.

Pre-lab Deliverable (14): Write a brief paragraph detailing any ideas or notes that came up in your communication with your partners.

Pre-lab Deliverable (15): On Gradescope, answer the multiple choice questions about reading Data Sheets.

8.4 Pre-Laboratory Assignment

Your assignment responses should be uploaded to the ECE 110L Laboratory Gradescope site by the assignment deadline.

EACH STUDENT must submit their own INDIVIDUAL assignment responses!

You will respond to the following:

1. Type the Duke Honor Code:
   - *"I have adhered to the Duke Community standard in completing this assignment."*
2. **Pre-lab Deliverable 1:** Write a brief paragraph describing your conversation with your partner(s) and any important notes you may have.
3. **Pre-lab Deliverable 2:** Answer the multiple choice questions on Gradescope.

Your assignment responses should be uploaded to the ECE 110L Laboratory Gradescope site by the assignment deadline.

EACH STUDENT must submit their own INDIVIDUAL assignment responses!

8.5 Experimental Exercise

1. Get together with your whole lab team to talk one last time about possible adjustments.
2. Run numerous trials before the trials starts. Prepare as a team to make the **"FULL SYSTEM DEMO"** go as flawlessly as possible.

8.6 IDC Full System Demonstration

See the IDC document for details. Your lab team will need to give the **"FULL SYSTEM DEMO"** during the first half hour of lab this week. Once you are finished, complete the deliverable below and prepare to give your Oral Defense.

IDC Deliverable (10): Update and submit a copy of your IDC code as a .ino file. This should be the final version of your code. Save this as a separate file. **Be sure your code is thoroughly commented.** This *does not mean every line needs a comment*, but there should be enough comments to make it clear what your code does. Don't forget about the standards we use and that are found in the Arduino Coding Guide.

IDC Deliverable (11): Record your "Group's IDC Score". Have this on hand to report as part of your submission.

8.7 Oral Defense

Following your **"FULL SYSTEM DEMO"** you will participate in the oral defense. The IDC Document provides information about the oral defense, but you are expected to:

- provide your TA(s) with an in depth knowledge of your code, your components, decisions made, and successes and failures your group worked through.
- make sure you can answer any questions about how your sensors work, including those for the sensing tasks and QTIs, as well as how your code is written and organized.

NOTE: You are allowed to have pictures of your 'bot with you during your oral defense, but not your actual 'bot. It is recommended that you take pictures of your 'bot before disassembling it. You will be required to disassemble your 'bot, organize your 'bot box, and take a picture of the 'bot box as a deliverable. See Figure 2.

IDC Deliverable (12): Participate in the Oral Defense. Provide TAs with an in-depth knowledge of your code, your components, decisions made, and successes and failures your group worked through. Be able to answer the very simple question about this on Gradescope.

8.8 Bot Box Organization and Cleanup

Before leaving, it is essential that you complete all of the following steps to ensure that your 'bot box is in the same, if not better, condition as when you began the IDC:

- Remove all components, hardware, labels, tools, and tape you added to your bot. This may include, but is not limited to: task sensors, QTIs, jumper wires, LEDs, resistors, buttons, switches, etc.
- Organize your group's wire kit/pack, making sure only wires are in it, and that it can close.
- Make sure that your 'bot still has the Arduino microcontroller attached to the CX-Shield, which is itself attached to the chassis. Additionally, the wheels and servos, and the battery pack (with batteries) is still attached to the chassis.

IDC Deliverable (13): Take a picture of your Bot Box after you have removed all components and hardware you added. In the picture, it should be clear that your bot has the CX-shield, Arduino, wheels and servos, and the battery pack (with batteries) fully attached. **Replace any missing screws or hardware**. It should also include an organized and cleaned out wire pack, your USB A-B cable, your AC/DC adapter (wall charger), 4 QTI sensors removed from the 'bot, your group's XBee module attached to it's SIP board, and your group's LCD. All other components, hardware, labels, tape, tools, and USB adapters are removed and properly dealt with.

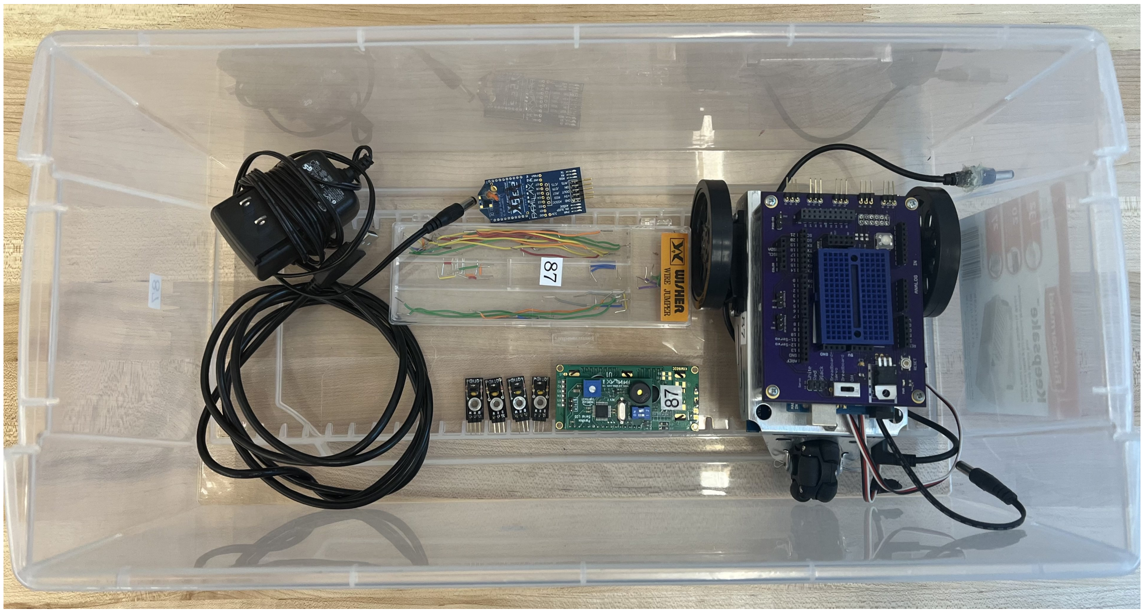


Figure 2. Example picture of Bot Box as it should look when returned.

8.9 Assignment

Your assignment responses should be uploaded to the ECE 110L Laboratory Gradescope site by the assignment deadline.

EACH STUDENT must submit their own INDIVIDUAL assignment responses!

You will respond to the following:

1. Type the Duke Honor Code:
   - *"I have adhered to the Duke Community standard in completing this assignment."*
2. **IDC Deliverable 1: Final Code used for Full System Demonstration** 8.6
   - Your group's FINAL code used for the Full System Demonstration should include everything the 'bot must do to complete the entirety of the IDC's required tasks, including any changes from the previous iteration and the EXPECTED performances based on the analysis and calculations.
   - *If the code is the same, comment this at the top of your code and then upload that version before you submit it.*
   - Be sure your code is thoroughly commented. This does not mean every line needs a comment, but there should be enough comments to make it clear what your code does.
3. **IDC Deliverable 2: "Group's IDC Score"** 8.6
   - Record your "Group's IDC Score" in the space provided.
4. **IDC Deliverable 3: Oral Defense Participation** 8.7

- Answer this yes or no question to confirm whether you completed the Oral Defense. If you provided details and answers to your TA's questions during the Oral Defense answer "Yes", else answer "No".

5. **IDC Deliverable 4: Bot Box Picture** 8.8

    - Upload a picture of your clean, organized Bot Box. It should be similar to what is shown in Figure 2.

Your assignment responses should be uploaded to the ECE 110L Laboratory Gradescope site by the assignment deadline.

EACH STUDENT must submit their own INDIVIDUAL assignment responses!

Answer the questions for the Lab 5.5 Online Assignment found on Gradescope.

Q1 Duke Honor Code

- Type the Duke Honor Code statement:"I have adhered to the Duke Community Standard in completing this assignment."

Q2 Arduino Sketch File 8.6

- Upload a .ino file of the code used to complete the "FULL SYSTEM DEMO".
- *Make sure you have the actual Arduino code, not just a screen capture or code in a google doc.*

Q3 Oral Defense Participation 8.7

- Answer this yes or no question to confirm that you completed the Oral Defense. If you provided details and answers to your TA's questions alongside your IDC partner(s) during the Oral Defense answer "Yes", else answer "No".

Q4 Bot Box Picture 8.8

- Upload a picture of your clean, organized Bot Box. It should be similar to what we see in Figure 2.

Q5 Team's IDC Grade 8.6

- Report the Overall Grade your group earned for the IDC to answer this question.

Your answers should be provided through this **ONLINE ASSIGNMENT** on the ECE 110L Laboratory **Gradescope** site by the assignment deadline. Each student must submit their own **INDIVIDUAL** assignment.

9 Appendix

9.1 Grading and Evaluation

For each checkpoint that your team completes in lab, you will receive **ten points**! The final two checkpoints require all five teams in your lab section to work together to complete, but coordinating early and often–starting, for example, with how to transmit signals consistently–will help make the final weeks more smooth. Thus, we encourage teams to communicate with each another throughout the IDC.

During the Full System Demo, you can receive up to **twenty-five points** (up to five points per trial). The entire demo will be scored on a 50 point scale, meaning that there are ten possible points per trial. For each trial within the full system demo, you must earn five of ten points in order to receive all five points toward your IDC grade. The scoring system is summarized below.

Task	FULL SYSTEM DEMO	50 Points Possible
Navigation & Sensing	**Did your bot correctly traverse the board to find an object?**	**15 Points Total**
Line Following	Path followed accurately using servos and QTIs	5 points
Sensing I	Object correctly detected in the first hashes	5 points
Sensing II	Object correctly detected in the last hash	5 points
Communication	**Did your bot correctly communicate information?**	**15 points total**
Transmit Data	Data corresponding to the correct state sent through XBee	5 points
Receive Data	Data from other robots succesfully received and read in through XBee	5 points
Display Data	LCD displays team's data values after they are all received	5 points
LED Indicators	**Did your LED indicators indicate what your bot was doing?**	**10 points total**
Sensing LEDs correct	Both the external and internal LEDs flash colors appropriately at hash marks	5 points
Comms LEDs correct	The external LED flashes colors appropriately based on XBee operation	5 points
Next Steps Determination	**Did your bot determine the correct next steps?**	**10 points total**
Correct celebration	Performs the correct task based on team state	5 points
Coordinated celebration	The celebrations are synchronized to the specified amount	5 points

9.2 The Oral Defense

After all five trials are completed, each student will meet with their TA for an “oral defense” – a debrief conversation of how the entire IDC went, what decisions were made, and how various components of the robot function. During this time, students should be prepared to respond to questions that their TA may have regarding the bot itself and the student’s design experience throughout the IDC. The oral defense is designed to evaluate students’ knowledge in four areas:

1. Arduino code function and organization
2. Sensor and device function
3. Engineering design decisions
4. Challenges faced and how they were addressed

In general, TAs will try to integrate these questions together and we encourage you to expand and elaborate on different aspects of the IDC, even if you’re not directly asked about them in a question! Additionally, students will not be expected to memorize or explain specific pieces of their code (see the Video Progress Report below), but may be asked why a specific data structure or function type was used.

To help prepare students for the Oral Defense, and to build their knowledge in the four areas identified above, the first four weeks of the IDC are paired with **“Oral Defense Prep”** activities. Each week, we will provide a list of questions that are associated with the four areas. Students will write a paragraph or two responding to them. These are low-stakes answers that will not be evaluated for accuracy – we just want you to get a glimpse at what the Oral Defense will be like and provide a place for you to get feedback on your answers to possible questions that we might ask! More details will be provided in the lab manuals for the IDC Weeks.

9.3 IDC Oral Defense Question Bank

In the Oral Defense you will demonstrate your learning across several key areas related to the IDC. Your responses in the Oral Defense should complement your work in the IDC Code Video—you are not expected to describe code in detail (rather describe your code’s functionality in a pseudocode manner). The Oral Defense should take at most five minutes, meaning that you will need to keep answers brief and to the point. Additionally, you will not be allowed external aids (e.g. computers).

The Oral Defense will focus on three areas:

1. Sensors (QTI and group-specific sensor):
   (a) How does the sensor physically transduce a signal?
   (b) What sorts of transduced signals are expected to be read in at the Arduino (if applicable)?
   (c) How does the Arduino convert the transduced signal into a measurement (if applicable)?
   (d) How can we use this measurement and/or signal to determine the presence or absence of an object (or line)?
2. How did the parameters of the IDC and/or your sensors influence design decisions?
   (a) Physical construction of the bot;
   (b) Code-based structures, such as communication protocols;
3. What challenges were faced during the IDC? How were they addressed?

You are expected to know the answers to the Oral Defense Prep questions and should review the discussion posts plus any feedback your TA has given you on your answers. Additionally, we have put together a bank of other questions that you may be asked on the Oral Defense below that you might want to review. The first question you will be asked is: *When your bot encounters an object at a hash mark, how does your bot determine that the object is present?*

Your answers to this question (and follow-up questions) should keep in mind the focus areas listed above. You will not need to reference specific code throughout this process.

Other questions you may want to consider when preparing for the oral defense include:

- How did you mount your object sensor to your bot and/or connect it to your Arduino?
- What challenges did you have installing your sensors, and what solutions did you implement?
- What kinds of pins (e.g. 5V, GND, Tx, Rx, etc.) does your object sensors use, and how is this similar/different to the pins used for your QTIs and your LCD?
- Based on the code you wrote and how you wired your object sensor, explain how it functions.
- How is your sensing integrated into your code (for example, what scenarios triggers the sensing task)?
- Explain how the QTI sensor uses an IR pulse and phototransitor to determine whether a line is present or not (also be able to briefly explain the importance of these components and the capacitor present in the QTI). How does this connect to what QTI stands for?
- Explain how your object sensor interacts with the environment to obtain a signal, if it does (e.g. does it send a pulse, use a physical law, etc.).
- Compare your object sensor to the QTI sensor everyone used. Explain some similarities and difference between how they work at a code and/or hardware level.
- Describe the system you are using to encode your bot's identity and sensing object information during the communication task. What are some benefits and drawbacks to this system?
- How did you determine and implement celebrations?
- How did you get your bot to stop at the last hash?
- How did your bot count hashes?
- Given more time, what is one improvement you would make to your bot?
- How does the Xbee's serial buffer function? Knowing how it works, what are some considerations you and your section should make in terms of sending data?
- What are some assumptions that you made when deciding how to mount QTIs on your bot?
- What are some of the design decisions your group made when mounting your sensor and what variables impacted these decisions?
- After you obtained sensor information from other bots, how did your bot determine which of the three celebrations to implement?
- How did your bot determine at what point to begin the celebration?
- How does your bot address not receiving a response from one or more bots in the section? If you do not implement this, what might you go about doing?

9.4 The Video Progress Report

Following the fifth week of the IDC, each group will collaboratively record a 5-7 minute video that outlines the major components of their code – including the major functional components (i.e. where in the code each task occurs and how the code executes that task, generally), how their code is structured/organized and why, and how the Arduino interacts with sensors or devices that are not integrated into the shield. This video need not be exhaustive, but should cover the parts of the code necessary to accomplish the four checkpoints. A rubric for the video will be released closer to the fifth week of the IDC

9.5 Checkpoints and Timeline

IDC Week 1

IDC Checkpoint 1 (10 points possible):

1. Navigating
2. Line following
3. Stopping at hashes

Oral Defense Prep 1 on Gradescope (5 points possible)

IDC Week 2

IDC Checkpoint 2 (10 points possible):

1. Sensing and transmitting
2. Using sensor to locate object
3. Sending score via the XBee

Oral Defense Prep 2 on Gradescope (5 points possible)

IDC Week 3

IDC Checkpoint 3 (10 points possible):

1. Receiving and displaying
2. Receive scores from other bots via XBee
3. Display data array on LCD

Oral Defense Prep 3 on Gradescope (5 points possible)

IDC Week 4

IDC Checkpoint 4 (10 points possible):

1. Analyzing and calculating
2. Identify "next steps" and implement outcome tasks

Oral Defense Prep 4 on Gradescope (5 points possible)

IDC Week 5

Complete unfinished IDC Checkpoints:

1. Design revision and finalizing
2. Prepare robots for trial runs!

Video progress report on Gradescope (12 points possible)

IDC Week 6

Full system demo (25 points possible)

1. The team will perform 5 trial runs in a row

Oral defense (12 points possible)

9.6 IDC Code Video Grading Criteria

Length Requirements

- Video must be between 5–7 minutes.

Code Explanation (6 pts)

- Students must explain how their code accomplishes IDC tasks.
- Award points ($+1$ per task) if the majority of the explanation is correct:
  - Line following
  - Presence of object
  - XBee communication
  - Display data
  - Determine celebration
  - Perform celebration

Object Sensor Interfacing (3 pts)

- $+0.5$ per signal: Students describe expected input and output signals for their sensor.
- $+1$: Students describe any libraries used to convert the transduced signal.
- $+1$: Students explain how code obtains the output signal from the input voltage signal.

Code Organization (2 pts)

- $+1$: Code makes use of functions for repeated tasks.
- $+1$: Code is commented clearly enough for readability without external help.

Video Quality (1 pt)

- Video must be comprehensible, professional, and within the time limit.

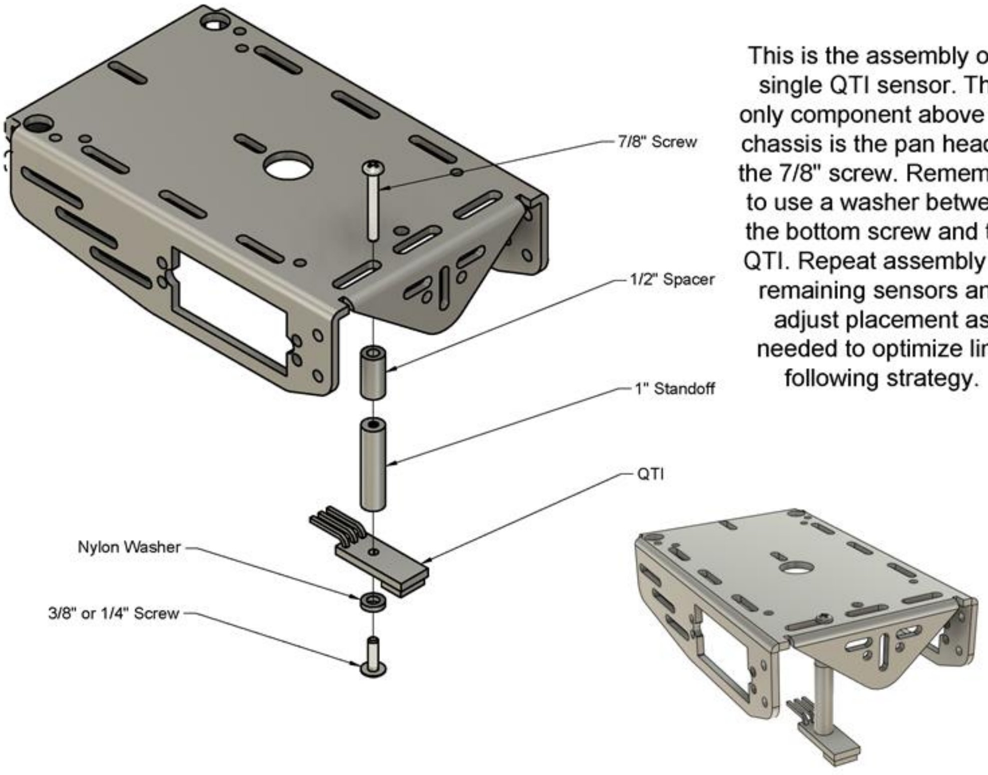
9.7 Spring 2026 IDC Evaluation Criteria

Full System Demo (50 points total)

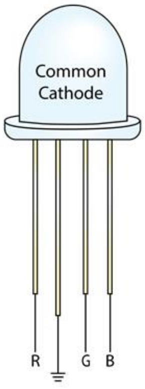
Grading: For each trial, students can earn up to 5 points toward their IDC grade (25 points total). The full 50 points are used to differentiate section performance. The following ten tasks award 1 point per trial for 5 trials:

1. **Line Following:** "Path followed accurately using servos and QTIs" Award point if the bot makes it to the final hash mark and stops. The bot should also stop at each hash mark. *No points awarded if the bot is touched at any point during the trial run and/or if the bot moves the sensed objects.* Confirm via observation.
2. **Sensing I:** "Correct readings determined/found at correct hash in the first stage" Award point if the bot correctly detects the object located in one of the 5 initial hashes. Confirm via the external RGB LED flashes and/or XBee transmission.
3. **Sensing II:** "Correct readings determined/found at correct hash in the second stage" Award point if the bot correctly detects if the object in the sixth hash. Confirm via the external RGB LED flashes and/or XBee transmission.
4. **Transmit Data:** "Data corresponding to correct hash sent through XBee" Award point if the bot transmits any data. Confirm via other bots' LCDs at the end of the trial, or via the TA bench XBee.
5. **Receive Data:** "Data from other robots received/read from XBee" Award point if the bot receives any parcel of data from another bot. Confirm via the LCD at the end of the trial, or via the RX pin on the XBee flashing during the trial.
6. **Display Data:** "LCD displays team's data values after they are all received" Award point if the bot displays data received from other bots in an array after all sending is complete. *If all bots do not successfully transmit data, do not take off points (points should still be awarded if the bot displays an array that only contains their sensed data).* Confirm via the LCD at the end of the trial.
7. **Sensing LEDs Correct:** "The red and green LEDs flash appropriately at the hash marks" Award point if an external RGB LED flashes red when at a hash mark without an object and green when at a hash mark with an object. Both hashes must be correctly indicated. Confirm via external RGB LED.
8. **Comms LEDs Correct:** "The blue and purple LEDs flash appropriately at the hash marks" Award point if an external RGB LED flashes blue when transmitting data and purple when receiving data. Both actions must be correctly indicated at some point. Confirm via external RGB LED.
9. **Correct Celebration:** "Bot performs the correct task based on board state" Award point if the bot performs the correct celebration based on the board state. *Do not award points for incorrect celebrations based on incorrect read-ins.* Confirm via the celebration performed.
10. **Coordinated Celebration:** "The celebrations are synchronized to the specified amount" Award point if all bots perform the correct celebration based on the board state within the specified synchronization standard. Confirm via Lab Manager.

9.8 QTI Mounting Illustration and External RGB LED Wiring



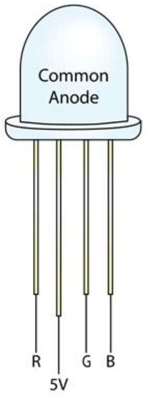
This is the assembly of a single QTI sensor. The only component above the chassis is the pan head of the 7/8" screw. Remember to use a washer between the bottom screw and the QTI. Repeat assembly for remaining sensors and adjust placement as needed to optimize line following strategy.



RGB LEDs work by way of passing voltage across diodes within the shell or casing. There is a pin for each of the three diodes, and a pin for either the cathode or the anode.

Although the arrangement of the R, G, and B pins can differ from these two examples. Often, one of the two middle pins will be longer than the others; it is the LED's common pin.

The common pin is either connected to ground (common cathode) or connected to a voltage source (common anode), such as a 5V pin on the CX-Shield.



Common Cathode:
In a common cathode configuration, all the cathodes (negative terminals) of the individual LEDs are connected together and brought out as a single negative pin (the common cathoode). That is then connected to a GND (ground) pin on the CX-Shield. To light up a specific color, you need to apply a high voltage to its corresponding anode (positive) pin. This is done by making its dedicated analog or digital pin on the CX-Shield have a high state.

Common Anode:
In a common anode configuration, all the anodes (positive terminals) of the individual LEDs are connected together and brought out as a single positive pin (the common anoode). To light up a specific color, you need to apply a low voltage to the corresponding cathode (negative) pin. This is done by making its dedicated analog or digital pin on the CX-Shield have a low state.