ME 218b Winter 2023: GIT Pushin'

Project Preview on March 2 from 1-7 pm.
Grading Period begins March 2, and runs through 11:59 pm on March 7.
Project Presentation on March 8 starting at 7:00 pm.

Goal:

The goal of this project is to provide you with an opportunity to apply your knowledge to solve an open-ended problem. The task is to design and build a machine that can autonomously navigate around the Gameplay Interaction Territory Hub (Hub, for short) and push Caster Operated Mechanical Masses In Transit (COMMITs) toward the Robotic Environments for Positioning & Orientation (REPOs) located at the end of the Battle-Ready Autonomous Navigation CHannels (BRANCHs).

Purpose:

The underlying purpose of this project is to give you some experience in integrating all that you have learned in ME218 as well as your prior courses. To gain this experience, you will design and implement an autonomous mobile robot that can compete in a game of speed, skill and strategy against machines constructed by other teams from the class.

Background:

Your task will be to develop, implement, and test a Dueling ***Ed***it Vehicle (DEV) that will participate in a game comprised of pushing COMMITs between the Local & Remote REPOs. As you know, when multiple DEVs push commits in the same BRANCH, merge conflicts are likely to occur. The goal is to resolve merge conflicts in your favor in order to gain control of the Hub for bragging rights across the Galaxy (prelims on March 7, with the tournament scheduled for March 8).

The Task:

During development, your DEVs will operate on the Hub located in the SPDL. Your DEVs will then compete against each other on the same Hub as it is moved to the staging area, the Atrium of Bldg. 550 (our classroom building), for public presentations.

> *There's no sense in being precise when you don't even know what you're talking about.*
>
> *John von Neumann*

Specifications

The Hub:

- ☐ The Hub is an approximately 244×244 cm area with exterior walls 8.25 cm tall. A top view is shown in Figure 1.
- ☐ The Hub is divided into 3 BRANCHs, Develop, Main & Hotfix, bounded by walls, to be used by Local and Remote as shown in Figure 1.
- ☐ At the ends of the Feature & Hotfix BRANCHs a modulated IR beacon will be mounted, with the emitters located 36 cm above the surface of the Hub. These are to allow the DEVs to fetch the identity of the REPOs. The modulation frequencies are shown in Table 1.
- ☐ A 2.54 cm wide black tape line runs through the center of each BRANCH.

The DEV:

- ☐ Your DEV must be a stand-alone entity, capable of meeting all specifications described in this document. Only SPDL supplied (or equivalent) battery power is allowed. No more than two batteries may be used to drive the motors that transfer force to the ground.
- ☐ Each DEV must be able to automatically determine if the DEV is working in the Local (red) or Remote (blue) REPOs.
- ☐ Each DEV must include a means to clearly indicate to the audience its team status.

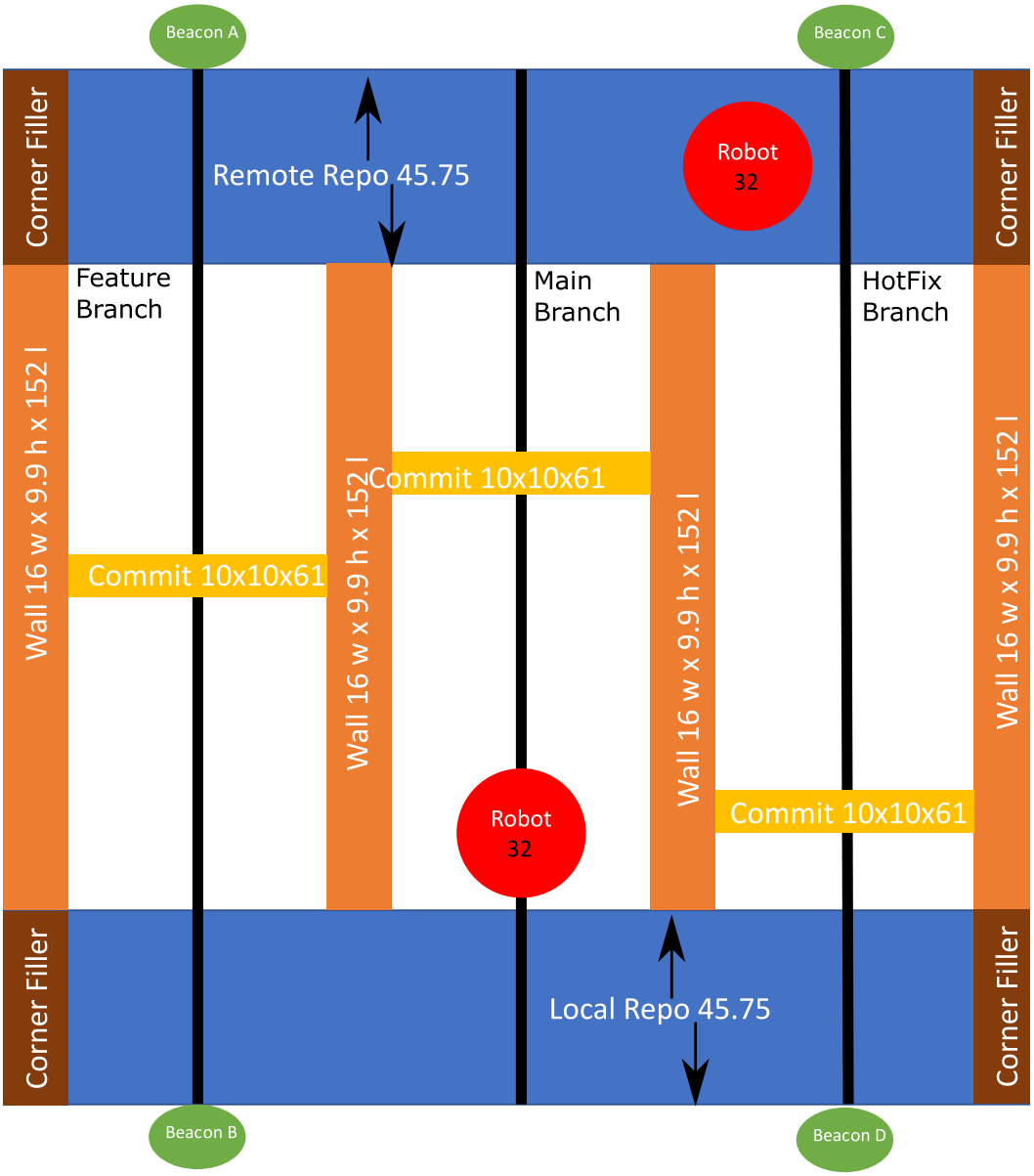

Figure 1: Overhead view of the Hub. All dimensions in cm and approximate. Refer to the actual field for exact dimensions.

Beacon	Frequency	Period
Beacon A	3333 Hz	300 µs
Beacon B	2000 Hz	500 µs
Beacon C	1427 Hz	700 µs
Beacon D	909 Hz	1100 µs

Table 1: Beacon Modulation Frequencies

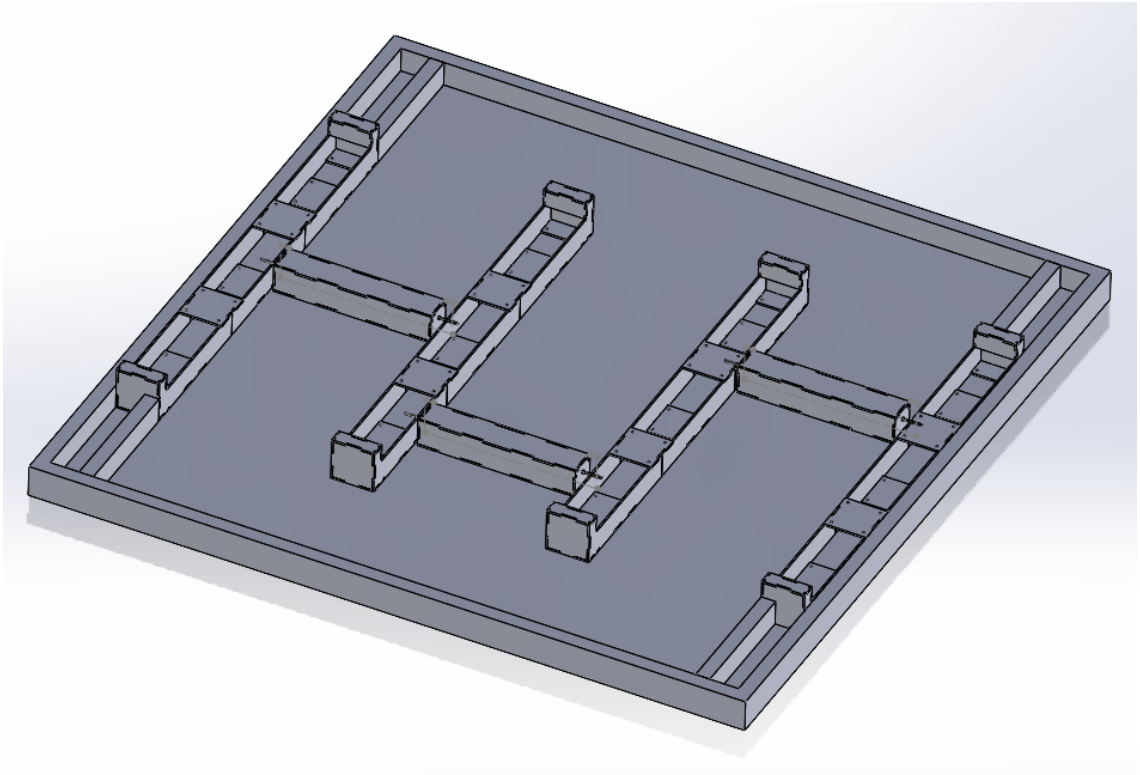

Figure 2: Isometric view of the Hub.

- ☐ DEVs must be autonomous and untethered.
- ☐ The only parts of the DEV that may ever touch the playing floor are wheels, ball casters, or slippery supports used to balance the DEV.
- ☐ Only the supplied motors may be used to drive anything that transfers force to the ground.
- ☐ At the beginning of the game, your DEV must be fully contained within a square, 32cm on a side, as projected into the field. The DEV must never expand in the plane of the Hub beyond the perimeter defined at the beginning of the Game. The height of the DEV must not exceed 32cm.
- ☐ Each DEV shall contain a network of processors consisting, at a minimum, of one SPI leader and one SPI follower device. How the necessary functionality is partitioned among the networked processors is up to the team.
- ☐ Each DEV must carry an easily accessible switch which shall cut power to the DEV in case of a software or hardware malfunction.
- ☐ The motor power supply for the DEV must be fused for protection in case of a hardware malfunction.
- ☐ Each DEV must be constructed as part of ME218b. It may not be based on a commercial or otherwise preexisting platform.
- ☐ The accounting department has limited you to an expenditure of **$150.00/team** for all materials and parts used in the construction of your project. Materials supplied to each team by SPDL or from the lab kit do not count against the limit. All other items count at their fair market value.
- ☐ Each DEV may have as many user accessible switches or buttons as the team desires but these switches or buttons may only be activated/modified while the DEV is stationary, fully within the bounds of the REPO Zone and an indicator is activated to indicate that the DEV is ready to receive input.

- ☐ Each DEV must provide a clearly visible, electro-mechanical indication of when it thinks that a Game is in progress. This indicator should be activated when the DEV determines the Game has started and be deactivated when the DEV thinks the Game has ended.
- ☐ There must be a bumper surrounding the perimeter of your DEV extending for at least 5 cm vertically, and must fully cover the region between 1.3 cm and 6.3 cm (± 0.3 cm) from the floor. The only exception is for small holes to allow for sensors.

The Game:

- ☐ The Game is a HEAD-to-HEAD race between two competing teams of DEVs operating from opposite ends of the BRANCHs. The objective is to move the COMMIT towards the opposing team's REPO so as to maximize the area of the BRANCHs on your side of the Hub.
- ☐ At the beginning of the Game, each team will place their DEV within their REPO region along the center line of the leftmost BRANCHs (refer to Fig. 1) and with an orientation designated by a member of the teaching staff.
- ☐ While the Game is in progress, if the DEV returns to the REPO Area and is fully stationary, a member of the team may interact with the DEV by pressing buttons or flipping switches. This process may be repeated as many times as desired within the time of the Game.
- ☐ The Game will begin when a member of the Teaching Staff announces that the Game has started. At this time, a member of each team is permitted to press the start button on their DEV. Any further interaction between team members and their DEVs is limited to pressing buttons or flipping switches and those interactions are only permitted while the DEV is stationary in the REPO Region and the input indicator is active.
- ☐ Each Game will last for 2 minutes, 18 seconds.
- ☐ When each DEV detects that the allotted time has expired, that DEV must cease moving and deactivate its game-in-progress indication.
- ☐ At the end of 2:18, the team whose COMMITs mark out the largest area along the BRANCHs is the winner.
- ☐ If both teams have an equal area at the end of the 2:18 regulation time, there will be a sudden-death round. The DEVs will be re-placed into the REPO zone with the DEVs and COMMITs in the same positions as at game start. The DEVs will be placed with a new, random orientation and, after the start command is given, the first team to push a COMMIT into the opposing REPO wins. If neither team completes the sudden death round, the game will be decided by an extremely dramatic coin flip.

Rules:

- ☐ No solderless breadboards are permitted in the final project.
- ☐ Intentional interference with the operation of another team's DEV is prohibited.
- ☐ Merge conflicts, pushing contests from opposite sides of a COMMIT, are allowed.
- ☐ Each DEV must start and remain in one piece during the round. Any locomotion of the DEV should cause all parts of the DEV to move.
- ☐ Your DEV may not **IN ANY WAY** alter the condition (e.g. mar the walls or floor) of the playing field.
- ☐ Your DEV may not **IN ANY WAY** damage, including jamming or toppling, the COMMITs.
- ☐ Intentional jamming of your opponent's senses is prohibited.
- ☐ All projects must respect the spirit of the rules. If your team is considering anything that may violate the spirit of the rules, you must consult a member of the teaching staff.

Safety:

- ☐ The DEV should be safe, both to the user and the spectators. The teaching staff reserves the right to disqualify any DEV considered unsafe. This also applies during testing, so keep the DEV velocity low enough so as not to cause problems.
- ☐ You should make a stand to support your DEV on the table for testing. The purpose of the stand is to prevent an errant DEV from running off of the table during testing.
- ☐ DEVs must be stable in the presence of a 15 m/s wind.
- ☐ No part of the DEV may become ballistic.
- ☐ All liquids, gels, and aerosols must be in three-ounce or smaller containers. All liquids, gels, and aerosols must be placed in a single, quart-size, zip-top, clear plastic bag. Each DEV can use only one, quart-size, zip-top, clear plastic bag.
- ☐ Red, Green, and Blue shells are prohibited. Any banana peels must stay within the confines of your DEV at all times.
- ☐ Any early celebrations will be penalized.
- ☐ Heads must remain attached at all times.
- ☐ Your DEV is not permitted to steal talent from any other DEVs.
- ☐ DEVs may alter the space-time continuum only during the public presentations.

Checkpoints

Design Review:

During the day on **February 14** we will conduct design reviews. Sign-ups for the hour-long slots for 3 teams will happen via a Google Sheet. Each group should prepare a few **simple** PowerPoint slides (scans of sketches are OK). **No code, no state diagrams, no circuits.** The focus should be on the overall design and how you are tackling what you think are the critical subsystems and how you will partition responsibility between the processors. You will present these to other members of the class, members of the teaching staff and coaches so that all may hear about your ideas and provide feedback and advice.

Many of these trees were my friends. Creatures I had known from nut or acorn.
Treebeard, preceding the battle of Isengard - The Lord of the Rings: The Two Towers

First Checkpoint:

On **2/16/23**, you will turn in a system block diagram, a set of KiCad schematics, textual descriptions and software design documentation (including a state chart) that describes the state of the design *at that point in time*. The designs need not be tested at this point, but must include designs to address all of the major subsystems. For your submission to GradeScope, create a single PDF document that includes the system block diagram, a KiCad schematic, your state charts, and a document describing, in words, your strategy for meeting the project requirements and identifying your robot's core functionality. Only one team member needs to submit your checkpoint documentation.

We can only see a short distance ahead, but we can see plenty there that needs to be done.
Alan Turing

Second Checkpoint:

On **2/22/23**, you must demonstrate communications between at least two PICs. Between them, these PICs must demonstrate the ability to control the motors of the locomotion system and the ability to detect the beacon. This demonstration may be a table-top demonstration and may be demonstrated while connected to a PC.

The key thing about all the world's big problems is that they have to be dealt with collectively. If we don't get collectively smarter, we're doomed.
Douglas Engelbart

Third Checkpoint:

On **2/27/23**, you must demonstrate an instance of your possibly tethered, motorized platform moving, though not necessarily with integrated software. Your platform must be able, starting from the start position on the Hub, to locate the beacon and push a COMMIT to the end of a BRANCH under coordinated control from multiple processors.

One accurate measurement is worth a thousand expert opinions.

Grace Hopper

Project Preview:

At the Project Preview on **3/2/23**, each team's DEVs must demonstrate, untethered and in an integrated physical form, under autonomous software control:

1. The ability to move around the Hub under software control.
2. The ability to locate the beacon(s) and identify your team.
3. The ability to push a COMMIT to the end of a BRANCH and return to your own REPO.

This will be tested by following the normal starting procedure for a Game, followed by the team's DEV performing the required maneuvers.

Talk is cheap. Show me the code.

Linus Torvalds

Grading:

The grading period will open on **3/2/23** and will remain open until 11:59 pm on **3/7/23**.

During a grading round, each team will be required to demonstrate a complete Game.

During the Game, your DEV must demonstrate in a completely integrated form all abilities detailed in the Project Preview specification.

Evaluation for grading purposes will only occur during these rounds. If your DEV fails at its first attempt to demonstrate its ability, it must then demonstrate the ability two times in succession at its next attempt. These increases continue after repeated failed attempts to a maximum of 4 required successive demonstrations.

There's a big difference between "impossible" and "hard to imagine." The first is about it; the second is about you!

Marvin Minsky

Public Presentation:

This will take place on **3/8/23** starting at **7:00 pm** in the Atrium of Building 550. (building in which our classroom is located)

Everything on the Internet is true.

Honest Abe Lincoln

Report:

Draft due on **3/13/23** by 4:00 pm. The final version with revisions is due by 5:00 pm on **3/17/23**.

Be the pull request you wish to see in the world.

github.com/careers

Evaluation

Performance Testing Procedures:

One or more team members will operate the DEV during the performance evaluation. A competition among the class' DEVs will take place after the performance evaluation.

Performance Evaluation:

Performance evaluation will take place twice during the project duration, at the Project Preview and at the Grading Session. Everyone will participate at this level.

The Competition:

On the night of the public presentations, a tournament will be held. **Performance during the tournament has no impact on your grade.**

Grading Criteria:

- ☐ **Concept (10 %)** This will be based on the technical merit of the design and coding for the machine. Included in this grade will be evaluation of the appropriateness of the solution, as well as innovative hardware, software and use of physical principles in the solution.
- ☐ **Implementation (15 %)** This will be based on the prototype displayed at the evaluation session. Included in this grade will be evaluation of the physical appearance of the prototype and quality of construction. We will concentrate heavily on craftsmanship and finished appearance.
- ☐ **Checkpoint Performance (10 %)** Based on demonstrating the required functionality at the **checkpoints**.
- ☐ **Preliminary Performance (10 %)** Based on the results of the performance testing during the **Project Preview**.
- ☐ **Performance (20 %)** Based on the results of the performance testing during the **Grading Session**.
- ☐ **Coaches' Evaluation (5 %)** Evaluation by your coach: did you make use of their input before the design review and during the course of the project.
- ☐ **Report (20 %)** This will be based on an evaluation of the written report. It will be judged on clarity of explanations, completeness and appropriateness of the documentation. The report should be in the form of a stand-alone web site and must include schematics, pseudo-code, state charts, header & code listings, dimensioned sketches/drawings showing relative scale, a complete Bill-of-Materials (BOM) for the project as well as a 1 page description of function and a "Gems of Wisdom for future generations of 218ers" page.

  To submit your report you must **enter the URL to your site into a Google sheet that will be made available for that purpose**. The only file types in your final report should be HTML (including style sheets if you choose), JPEG or other viewable image files and PDF files. Schematics should be PDF files, not bitmaps (PNG, JPEG, GIF, etc.). A reasonable resolution bitmap place-holder with a link to a PDF is the best solution to readability. **Do not simply place a link to the PDF of the schematic without a viewable preview on the web page.** Do not include `.doc`, `.docx`, `.xls`, `.xlsx` or other files that require opening a separate application outside of the browser. Your site should be fully functional on both desktop and tablet browsers. **Do not embed video files directly** into your site. If you want to include video, link to YouTube or other video sharing site.

  It is critical that the URL of your report be in the Google sheet on time so that the peer reviewing team will have an adequate opportunity to review it before class the following day. Final versions of the reports, incorporating the review comments are due (again, as a URL in the Google Sheet; update the URL if it changed) by 5:00 pm on 3/17/23. Make sure to test all of your links before submitting. If we can't simply open the link and read it on our machines, then we can't grade it.
- ☐ **Report Review (10 %)** These points will be awarded based on the thoroughness of your review of your partner team's report. Read the explanations, do they make sense? Review the circuits, do they look like they should work? Could this DEV realistically be built for $150? If, during grading, we find things that don't make sense or circuits that won't work we will consult your review. If the review caught them, then the team will lose points on their report. If the reviewers missed it, then the reviewers will lose points for their review. The report review should submitted be in the form of a word document that you place into one of your team members folders by 4:00 pm on 3/14/23.
- ☐ **Housekeeping** Based on the timely return of SPDL components, cleanliness of group workstations as well as the overall cleanliness of the lab. No grades will be recorded for teams that have not returned or

paid replacement costs for the items borrowed from the SPDL, including but not limited to the oscilloscope, power supplies, logic analyzer, tools, etc.

- ☐ **Peer Reviews** Completing the peer review on CATME is a **required** component of the project. These reviews are not optional and the project work will be considered incomplete unless the reviews are completed by the due dates.

Team Organization

While each member of your team has principal design responsibility for a specific functional area, I believe that turning these team members into dedicated specialists will be a mistake in the long run. I have heard from many 218 alumni who did this and reported that they were sad that they had done this because they didn't get, for example, communications experience. I would like to encourage you to remember that, first and foremost, **the purpose of the project is to enhance your learning of the material**. An organization that deeply involves all of the team members in the details of the design, implementation and debugging of all subsystems will not only provide a better learning experience, it will also prevent you from getting hung up during the integration and testing phase because the "expert" on that subsystem is not available.

Mechanical Design and Robustness

Your machine must be rugged enough to survive your testing.

While the emphasis in the lecture has concentrated on the electronics and software, don't forget the mechanical aspect. Historically, project failures are often due to poor mechanical design or implementation. Pay attention to craftsmanship, and put thought into how your design supports all of the loads your robot will be subject to—not just when it's operating as intended, but also when it receives bumps from other robots that may also be testing on the field.

While we have focused largely on software implementation details in class, keep in mind that although computers are deterministic, the real world is not[1]. Make sure your software is built not just to handle what you hope will happen, but also everything you think might happen that could cause you problems.

This year's project has more "smart" subsystems than other projects in the past. When integrating several subsystems, although it may seem (and feel) slower, it is absolutely worth your time to make sure each subsystem is as bug-free as you can get it prior to integration. Fast is slow and slow is fast.

Preventing Disaster

It is unlikely, even given the advice in the paragraph above, that your DEV will be robust to a fall from the table-top to the floor. To avoid the possibility of that happening, you should create a stand/platform for your DEV that it can sit on with its wheels not touching the table-top. With this stand in place and your robot perched upon it whenever it is on the bench-top, even if your code or hardware goes haywire and starts the wheels spinning unexpectedly , those spinning wheels will not drive your DEV off the bench-top.

Resources

Websites:

SparkFun	Seeed Studio	Jameco	Mouser
Newark	Ponoko	Adafruit	Hackaday
DigiKey	McMaster-Carr	HobbyKing	ServoCity
Pololu			

You may also find PlantUML and PlantText helpful for creating message sequence diagrams or system diagrams.

[1] Well, at small length scales. But in this case, you don't have enough information on the details of your robot's interaction with the world to treat it as deterministic, so don't.

Local Stores:

Anchor Electronics in Santa Clara
Jameco in Belmont
TAP Plastics in San Mateo

Gems of Wisdom:

Be sure to check out The ME218 Archive for guidance from past generations.

Team Names:

Team names are a tradition in ME218b. For some very Tongue-in-Cheek inspiration, take a look at git-cheat-sheet-education.