EE 1301 Final Project Specifications

Introduction: This document outlines the requirements and guidelines for your EE 1301 final project. Please read all points carefully, as they have been updated to reflect new expectations. The final project is an opportunity to demonstrate your creativity, technical skills, and effective use of modern tools. The following sections detail the key guidelines for the project.

Team Composition

You may choose to work individually or in a team of up to four students. Make sure all team members contribute significantly if you work in a group.

Project Scope and Visual Component

You are **encouraged (but not required)** to incorporate the Particle Photon (our IoT development board) or other hardware into your project. It is acceptable to pursue a project that is entirely software-based without any physical sensors or actuators. **However, every project must have a physical or visual component to showcase results.** If your project does not involve IoT/robotic hardware, then it must include a visual display or on-screen output as part of the demonstration (for example, a website, a graphical visualization, a GUI, or a game screen).

Originality and Ambition

Originality: Your project idea should include a unique twist or feature that sets it apart. While many projects will have similarities to existing ones, you need to add *something* original. If a quick search reveals a nearly identical project already exists, expect a lower grade for lack of originality. Take an existing idea further or combine ideas in a new way to make it your own.

Ambition: You are encouraged to be ambitious and attempt something bold or challenging. You will **not** be automatically penalized for attempting a difficult project that doesn't fully work out. In fact, trying something difficult is encouraged. However, you **must** have something functional to demonstrate at the end (even if it's only a portion of your original goal), and you must write a substantial report. In your report, explain what progress you made, the challenges you faced, and how you would address those challenges with more time. In other words, a partially successful ambitious project with a thorough explanation will earn a good grade, whereas an unoriginal or trivial project that works perfectly will not.

Use of AI in Coding (Required)

You are **required** to use an AI tool to generate part of your code for this project. You can generate all the code with AI if you choose, or only use AI for certain functions or modules. This can be done with code generation assistants or AI chatbots (for example, using ChatGPT or GitHub Copilot to help write a function).

In your final report, you **must include a description of which parts of the code the AI helped generate**. Be specific about what was AI-generated vs. what you wrote manually. Include the exact prompts or queries you used to obtain that code from the AI. This documentation demonstrates that you learned how to effectively leverage AI in coding.

Show your process: Include in your report examples of AI prompts and outcomes. For instance, provide **three prompts that initially generated code which didn't work or didn't do what you intended**, and explain how you revised your prompts to get correct results. Also include **three prompts that successfully generated substantial chunks of functional code** for your project. (Describe what the AI-produced code does and how you integrated it.) Do **not** dump a complete list of every prompt you used – just highlight a few key examples as described above.

By documenting these attempts, you'll show how you used AI to troubleshoot and develop your code. Remember that using AI is meant to enhance your coding ability and not replace your understanding. You are responsible for verifying that the code works and meets the project requirements.

Use of AI in Writing the Report (Required)

You are also **required** to use AI assistance to help write certain parts of your project report. (You may use it to draft the entire report if you wish, but **you must carefully edit and verify the content**.) Specifically, you should **first prepare bullet-point notes or an outline** for the following sections of your report, and then use an AI writing tool to expand those notes into well-written paragraphs:

- **Background:** Provide context or theory behind your project.
- **Related Work:** Discuss similar projects or research that relates to your idea.
- **Applications:** Describe potential applications or future extensions of your project.

For these sections, take advantage of AI writing tools to generate polished text from your outline. **In your final report, include the bullet points you drafted and the prompts you gave to the AI**, as well as relevant excerpts of the AI-generated content. Clearly mark or annotate the portions of the text that had AI assistance so we can see how you used the tool. The goal is for you to practice using AI as a writing assistant and to reflect on its contributions.

Important: You must read and **verify all text that the AI generates.** Do not blindly trust the AI's output. You will be penalized if you include AI-generated text that is factually incorrect or irrelevant to your project (for example, made-up technical details or nonexistent references). The intent is to teach you how to use AI as a research and writing resource while maintaining a critical eye. A proper workflow would be: use AI (and traditional web searches) to gather information, formulate a clear outline, prompt the AI to write text based on that outline (with explicit instructions or constraints you provide), then carefully read and correct what it produces. Repeat this process until the AI-assisted sections are accurate and well-written. By following this method, you can harness AI to help write your report while ensuring the content remains correct and specific to your project.

Project Assignments and Timeline

The project will consist of several deliverables throughout the semester. Below is an overview of each deliverable, including due dates and grading weight:

- **Project Proposal – Initial:** *Due October 23* (10% of project grade) – about 1 page.
- **Project Proposal – Update:** *Due November 7* (30% of project grade) – up to 3 pages.
- **Project Showcase:** *December 9* (20% of project grade) – in-person demonstration.
- **Final Written Report:** *Due December 9* (40% of project grade) – approximately 5 pages of text (and up to 15 pages of appendices including code excerpts, schematics, etc.).

Each component will be described in more detail below. All submissions will be via Canvas. Plan your work so that you can meet these deadlines; late submissions may not receive full credit.

Project Proposal – Initial (due Oct 23)

The initial proposal is a formal written document (approximately one page) that briefly describes what you plan to do for your project. This proposal is not binding – we understand your project

may evolve – but it should give a clear direction. Include the following information in your proposal:

- **Project Title:** An interesting, descriptive title for your project. (Be playful!)

- **Team Members:** The names of your group members (or just yourself if solo) and your University of Minnesota email addresses.

- **Problem Description:** A paragraph describing the problem you intend to solve or the need your project addresses. Why is this project interesting or worthwhile? Or simply cool?

- **Proposed Solution:** A paragraph describing the device or software you plan to create. Outline the key features and how they solve the problem.

- **Sensors/Actuators:** List any sensors and actuators you plan to use, and what they will do in the project.

- **Software component:** Summarize what software language(s) you will use.

Keep the proposal concise but informative – one p[age. It should convince the reader that you have a feasible project idea with a clear goal. **Submit your written proposal via Canvas** by the due date.

Project Proposal – Update (due Nov 7)

About two weeks after the initial proposal, you must submit a Proposal Update (up to 3 pages). This is essentially a more detailed plan, incorporating any changes since the initial proposal. **Each team will also have an in-person check-in meeting** with one of the TAs around this time to discuss your progress (the TAs will schedule this with you).

Your Proposal Update should include:

- **Updated Project Summary:** An updated description of your project, including any changes to the scope or goals since the initial proposal. Refine the problem statement and solution overview if needed.

- **Detailed Components:** An updated list of all sensors, actuators, and any cloud or internet features you plan to implement. Mention any new parts or services you decided to use.

- **Operational Diagram:** At least one visualization (e.g. a block diagram, flowchart, or storyboard) showing how your project will operate. This could be a flow diagram of the system, a user interface sketch, or any visual that helps explain the project's working.

- **Hardware Schematic:** A preliminary circuit schematic or diagram showing how you will wire your components (sensors, actuators, Photon, etc.). This does not have to be final, but should indicate you've thought through the connections.

- **Progress and Code Snippets:** Describe the progress you have made so far. Include basic functional code snippets for each major component (each sensor, actuator, or cloud feature). These code snippets can be simple test code demonstrating that you know how to read from a sensor or activate an actuator. They do not all have to work together yet, but by now you should have each piece at least individually working or a clear plan for it. You can include code inline or in an appendix. (The first 3 pages of the proposal update should stand on their own, with additional diagrams or code in appendices as needed.)

This update will be more substantial than the initial proposal. It should show that you have moved beyond planning into early implementation. The in-person check-in with a TA will be an opportunity to get feedback or help on any challenges you're facing. **Note:** The Proposal Update document is submitted on Canvas, and you should be prepared to discuss it during your check-in meeting.

IoT Project Showcase (Dec 9)

On December 8, we will hold an **IoT Project Showcase** in the **McNamara Atrium** (McNamara Alumni Center). You will present your project to the public, including course instructors, TAs, fellow classmates, and other attendees (guest judges, industry visitors, and anyone who stops by). This is your chance to show off your work—enjoy it!

At the showcase, you must **demonstrate your functional project**. If your project involves hardware, bring your device (and any required accessories) and be ready to run it live. If it's a software project, have it running on a laptop or display. You should be able to explain what your project does, what problem it solves, and how it works to people who visit your table. Each team will have a small space to set up a poster or laptop and the project itself. All team members should attend if possible and be ready to answer questions. Dress is casual, but remember you are presenting to judges and possibly the public, so professionalism is encouraged.

The showcase is a graded component, so make sure your project is as functional as possible by this date. Even if not all features are complete, have a **working demo** of some core functionality. We (and the judges) want to see something in action. Most importantly, have fun demonstrating what you built!

Final Written Report (due Dec 9)

The final report is a formal document that describes **what you actually did**, and more importantly **how you did it**. While the proposal and update outlined your plans, the final report

discusses the completed project (or the final state of it) in detail. This report is due by the end of the day on Dec 8 (the day after the last class day).

Length and Format: The report should contain around *five pages of written text* (not counting images, diagrams, code listings, etc.). It should also include up to 15 pages of code excerpts, schematics, and any appendices. Include all important images and diagrams to help explain your project, but focus on clarity and conciseness. The report must be submitted as a **single PDF file**. Use a readable font for code (fixed-width font, minimal line wrapping, and syntax highlighting if possible).

Your final written report should include the following sections:

- **Title Page:** Include an interesting project title, the names of your group members, and your contact information (student emails).

- **Project Description (approximately 4 pages):** Describe the project in detail, including:

  - **Problem Addressed:** What problem or need does your project tackle? Why is it important or interesting?

  - **Background:** Provide context or theory behind your project. (Written with AI – see above.)

  - **Related Work:** Discuss similar projects or research that relates to your idea. (Written with AI – see above.)

  - **Solution Overview:** How does your project solve the problem? Give an overview of what you built. If you built a device, describe its overall functionality. If it's software, describe what it does for the user.

  - **Components:** Describe each sensor and actuator in your project and their role (don't just list them—explain how each was used). Also describe how the devices/services are connected to the internet and what functionality that enables. For example, explain if you used Wi-Fi to retrieve data (weather, calendar info, etc.) or to send notifications, etc.

  - **Applications:** Describe potential applications or future extensions of your project. (Written with AI – see above.)

- **Code Design (approximately 1 page):** Explain how your program code is structured and how it works. Rather than listing every variable or function, create a narrative that breaks the code into logical pieces and describes the interaction between those pieces. For instance,

explain the overall flow (setup, loop, interrupts, cloud functions, etc.), and then dive into at least one **interesting or technically difficult part** of your code in detail. You might highlight a complex function or algorithm you implemented. You can include small code excerpts here to illustrate key sections (larger code listings should go in an appendix). Discuss any significant challenges in coding and how you overcame them (especially highlight if AI tools helped with those parts).

- **Results and Reflection:** Describe what you achieved and the current status of the project. Did it fully solve the problem? What **functional parts** are working as intended, and were there any parts that did not work as hoped? Include any testing you did. If you collected data or have results (logs, measurements, user feedback), summarize them. Reflect on what you learned from the project and what you would do differently or improve if you had more time. This is also a good place to comment on how effective the AI tools were in your development process (both coding and writing).

Additionally, include the following in your PDF (most of these can go in appendices after the main text):

- **Electrical Schematic(s):** Clear diagrams showing how everything was wired to your Particle Photon or other hardware. Anyone reading the schematic should be able to understand how to replicate your circuit. Label the major components.

- **Code Listing:** An excerpt of the code. This should be formatted for readability (proper indentation, line breaks, and ideally syntax highlighting). You can include all your code, if it fits, or just excerpts. (Suggestion: use AI to create *annotated* excerpts).

- **AI Usage Documentation:** As described above, include the bullet points, prompts, and AI-generated text excerpts for the Background, Related Work, and Applications sections of your report, as well as the examples of AI-generated code prompts/results. Make sure these are clearly labeled and placed (for example, in an appendix or highlighted in the relevant sections) so that it's clear how you used AI tools in your project.

Finally, remember that the **final report must be your own work** (with proper use of AI as allowed). All team members should contribute to writing and editing it. We will grade the report on how well it covers the required topics, the clarity of explanations, the depth of insight into your design and results, and the organization/professionalism of the document.

Example Project Ideas

To help you start brainstorming, here are some examples of project ideas. These are just suggestions – you are free to come up with your own original idea (in fact, we encourage it!). The key is to be creative and think about what would be fun, useful, or interesting to build. Each example lists possible inputs, outputs, and connections to give you a sense of scope:

Smart Greenhouse Monitor

Inputs: Temperature and humidity sensor, soil moisture sensor, light sensor (for ambient light levels).

Outputs: Water pump or servo controlling irrigation, small fan or vent actuator to regulate temperature, maybe an LCD or LED indicator for status.

Connection: Logs data to the cloud and fetches weather forecasts. For example, if rain is predicted, it might delay watering. You could monitor the greenhouse conditions remotely via a web dashboard.

Automated Window Blinds

Inputs: Light sensor (to detect sunlight), optional temperature sensor or clock/time input from the internet.

Outputs: Servo motors attached to window blinds to open or close them.

Connection: Fetches sunrise/sunset times or weather info from the internet. Blinds could open in the morning automatically and close during hot afternoons or at dusk. This project can help regulate indoor temperature and lighting automatically.

Smart Pet Feeder

Inputs: Real-time clock or schedule (feeding times), optional sensor to detect pet presence (motion sensor or weight sensor under food bowl).

Outputs: Servo or motor controlling a dispenser that releases pet food at scheduled times, indicator LED or buzzer that signals feeding time.

Connection: Allows remote monitoring and control via a web interface or mobile app. For example, sends an alert to your phone when the pet has been fed (or if food supply is low), and lets you trigger a manual feeding from your phone.

Mailbox / Package Alert System

Inputs: Magnetic contact switch or photo sensor in your mailbox (to detect door opening), or a pressure sensor on a doormat (to detect a delivered package).

Outputs: An LED indicator or buzzer in your house that notifies you of new mail, and/or an LCD display that shows “Mail has arrived!”. Could also activate a webcam snapshot.

Connection: Sends a notification to your email or phone (via an app or text) whenever mail or a package is delivered. This way, you know in real-time when something arrives at your doorstep.

Smart Alarm Clock

Inputs: Real-time clock and your online calendar events; optionally a microphone or motion sensor to detect when you're awake (or a button to snooze).

Outputs: A speaker or buzzer to play an alarm sound, a smart light or LED strip that gradually brightens, or even a servo that opens window blinds to let light in.

Connection: Checks your Google Calendar (or other calendar) for the next day's schedule and sets the alarm accordingly. For example, if you have an early meeting, it will ring earlier. It could also fetch weather or traffic data to adjust the wake-up time slightly (to account for heavy traffic, etc.). The device might send a summary of your first tasks of the day to a small display when you wake up.

IoT Arcade Game with Leaderboard

Inputs: A joystick (and possibly a couple of push buttons) to control the game.

Outputs: An 8x8 LED matrix or small LCD screen to display simple game graphics (for example, a Snake game or Pong). Maybe a buzzer for sound effects.

Connection: The game scores are uploaded to a cloud service or Google Sheets. This way, you can create a global leaderboard accessible via a web page. Friends or classmates could also play your game and compete for the high score. This project combines fun hardware control with an internet-connected score tracking feature.

Automated Pet Entertainer

Inputs: Motion sensor or webcam (to detect when your pet is nearby), or a timer that triggers play sessions periodically.

Outputs: A laser pointer or toy controlled by servo motors that moves in a playful pattern to entertain a cat or dog. Alternatively, a motorized fetching machine for dogs.

Connection: Can send a notification to your phone when a play session starts, and could be remotely activated. For instance, you could start a play session from work via an app. This project brings together sensors, actuators, and remote interaction for pet care.

Feel free to draw inspiration from these ideas and adapt them. The best projects often come from your own interests — think of something you wish existed or a problem you'd like to solve, and build a prototype for it!

"Group Kit" for the Final Project

We have a **"Group Kit" of parts** available for final projects. The group kit consists of various components (each roughly $5–$30 in value) that can be **borrowed** for your project. These items should be returned after the IoT Showcase. Examples of parts in this kit include:

- Temperature and humidity sensor modules (e.g., DHT11/DHT22)
- An analog joystick (X-Y thumbstick, like a small game controller stick)
- An 8×8 RGB LED matrix display
- Soil moisture sensors
- Small servo motors and motor drivers
- **(and more)**: relays, water pumps, LED strips, keypads, Bluetooth modules, etc.

You should familiarize yourself with the extended list of items in this kit so you know what is available and possible to use. The list of parts in the "Group Kit" is at: https://z.umn.edu/EE1301_GroupKit. The list of parts above includes tutorials and helpful web links to help you get started, and some of the parts have "Device Descriptions." If your project requires a special component not in the kit or your personal kit, talk to the instructor – we may be able to help you obtain it.

Good luck, and have fun with your project! This final project is a chance to be inventive and apply everything you've learned. We are excited to see what you create. Remember to start early, iterate often, and make use of the resources available (TAs, instructors, lab equipment, and AI tools). With careful planning and enthusiastic effort, you'll build something you can be proud of by the end of the semester. Enjoy the process of learning and making!