Physics 371

Project Design and Execution in a Physics Context:

Design Like a Physicist

Syllabus and Introductory Material
Spring 2023 course

George Gollin

Riccardo Longo

University of Illinois at Urbana-Champaign

Physics 371 – Spring 2023

George Gollin
Riccardo Longo
University of Illinois at Urbana-Champaign

Fridays, 1 pm – 5 pm; 3 credit hours.

Syllabus and Introductory Material

Brief description of the course

"Becoming the fearless toolsmith: you will address a real-world problem with your physicist's insight and the tools of electrical and mechanical engineers. There will be IDEs and PCBs and 3-D printers, and the remarkable experience of working collaboratively alongside fellow students and course staff."

Introduction: project physics

Some years ago Carl Wieman won the Nobel Prize for creating a Bose-Einstein condensate in a dilute cloud of 2,000 atoms. At the time he was a professor at the University of Colorado, and had noticed that his physics students appeared to undergo a dramatic transition during the first year of graduate school. As undergraduates, they would attend lecture-based classes and master course content by listening to their professors and slogging through weekly problem sets. (You all know what this is like!) By the end of the semester, most of the class would understand most of the material but would find it difficult to integrate it into a coherent picture of, say, classical electrodynamics. And a semester after a course had ended, most students would not have retained their mastery of the topic. They would find it difficult to apply the material in, say, a lab course. But after a year of graduate school—during which students would work on difficult material without the distracting edge effects of 50-minute class periods—their competence at navigating confusing subjects and difficult problems would increase enormously.

Wieman thought that teaching physics to undergraduates in a manner that more closely resembled graduate education might be beneficial. He began to explore project-based courses, in which students would learn physics by mastering what they needed to complete tasks that were more like research projects than was usually true in undergraduate instruction. The results were dramatic.

You've already had some experience with this instructional mode if you've taken Physics 298owl from Prof. Gollin. It's different from fighting to stay awake for an hour in a lecture, then sifting through the wreckage to extract what you need to do the homework assignment!

You will be performing the one-semester analogue of a PhD research thesis: defining a measurement to be performed, designing and building an instrument that might be capable of recording data necessary for the measurement, testing your device, doing the fieldwork to record valid data, then analyzing the data to form supportable, reproducible conclusions. If all goes well, you'll find this so captivating that putting your work aside to attend to your other academic obligations will be hard. I suspect it is this strong engagement with a project that drives the transition from an undergraduate level of skill to the expert mastery typical of graduate researchers.

Your device will comprise an embedded processor—many of you will use a Microchip Technology Inc. ATmega2560 microcontroller on an Arduino Mega 2560 board—interfaced to a suite of sensors built onto small "breakout" printed circuit boards. The Arduino's USB interface will allow you to download your (compiled) programs to the microcontroller and communicate with the processor through a serial interface.

After selecting your research partners and choosing a project, you'll begin assembling your instrument on a breadboard. You will develop the programs necessary to drive the various sensors, integrate them into a data acquisition system of your own design, build a more robust version of your device on a printed circuit board, 3D-print a case for it, and venture out into the world to do field work and record data. You will analyze your data, draw (and justify) conclusions, and document (and present) your findings. In the interest of efficiency, I encourage you to use in your design as much public-domain material as you are able to find. There is, in this course, no reason to reinvent the wheel.

Note that the course is NOT available for concurrent enrollment by high school students.

There is no text for this course. Dedicated resources can be provided upon request to address specific issues. Such requests should be placed during the weekly meetings with the instructor and the TAs.

Public health considerations

You will need to go about your collaborative work in a manner that complies with any University policies concerning public health and safety. For example, during the Covid-19 pandemic, you are/were expected to communicate with other project group members and course staff while properly wearing (non-vented, preferably N95) masks and maintaining CDC-recommended distances.

In greater detail, the University's policies concerning covid-19 are described here: https://covid19.illinois.edu/on-campus/on-campus-students/. Note that the penalties for violating the University's public health policies are substantial.

Prerequisites and corequisites

You must already know how to program. If you've learned to code in Python or C/C++, or Java, or some other language, you'll do fine. A B- or better grade in CS 101, CS 125, or Physics 298owl is a suitable prerequisite. It's also fine if you've learned on your own. If you've never programmed before, consider delaying enrollment in Physics 371 until after you've done some coding. Please note that a few "intensive seminars" about data analysis tools and techniques will be offered throughout the 2023 Spring course. Still, these mini-lectures (<= 1 h each) are meant to provide you with an overview of widely available tools and techniques you can explore afterwards on your own and eventually use within your project. In other words, do not expect to learn the coding skills needed to take this course from these seminars.

You must have a basic working knowledge of introductory physics at the level of Physics 211 and Physics 212. More is better, though not necessary. There are no corequisites for the course.

Physics 371 has minimal overlap with other courses in Physics and other departments' offerings. However, some of the skills developed in other courses (for example, familiarity with high-level programming languages) will be important to your work in this course.

We are not building robots

Physics 371 is not a course in robot building. That would be an engineer thing; we are physicists, not engineers. We will tackle measurements that—if they prove feasible—might make our corner of the world a little bit better. If we ***did*** build a robot, it would be to accomplish a significant end, for example recognizing the onset of a potentially catastrophic fall by an elderly person.

In Physics 371 you'll construct a hand-held device loaded with inexpensive sensors that are interrogated by a microcontroller—a small computer larded with additional features such as timers and analog-to-digital converters—and write the data acquisition software necessary to perform the measurements associated with your project. You'll assemble a prototype on a breadboard, construct a final (electrically equivalent) version on a printed circuit board, use a 3D printer to build a case for it, do fieldwork, and then write analysis code to understand what conclusions can be drawn from your data. You'll write a report presenting your results and justifying your conclusions, then publish it to the course website.

We will loan you the parts and tools necessary to construct the prototype, and will expect you to return these at the end of the course. But we will give you what you need to build the PCB version, and let you keep some of it at the end of the term. Please note that, if you withdraw from the course we'll want you to return everything we've given you.

The intellectual tradition in physics is for researchers to build their own instruments (buying off-the-shelf parts when available), ultimately creating sophisticated devices to perform the measurements that will tell us about the physics we are researching. It is not like this in all fields; But, in this course, you'll be following the physics tradition, and you will be working in (intellectually) close collaboration with two or three other students. The initial roaster of the class for Spring 2023 is comprised of 36 students. For this reason, at the beginning of the course you will be asked to organize in 9 groups of 4.

Some possible projects

Some of the projects are probably best imagined as feasibility studies that might inform the design of a more definitive future measurement. We will see how it goes!

Here are some that I have in mind. You are free to suggest other possibilities, though I reserve the right to veto anything that I feel is too difficult or too expensive.

- Radio-linked vector anemometers (Prof. Gollin's invention!): could a sound engineer use these to correct (in real-time) for wind-induced phase errors between towers of speakers in a large outdoor concert venue?
- How many beetles are in a cornfield? Or in traps disseminated through them? Or on single cobs? Prof. Nick Seiter and his collaborators at ACES are interested in these questions. Maybe we can help them!
    - Using mockup targets (cobs, traps, leaves etc..) with fake beetles in the lab, we can try to develop a device capable of acquiring images of the target (also surveying the neighboring environment) and counting the number of beetles on the target via pattern recognition techniques. If this part of the project is accomplished successfully, we can think about how to acquire data in fields
- Mapping crowd flow across Loomis: can we instrument a device to measure the macroscopic flux of people through the (labyrintic) Loomis corridors? What about typical room's typical hourly occupancy? This may be useful for crowd control and optimization of social distancing measures!
    - What about pairing a movement sensor with associated picture acquisitions at ground level, to then count # of shoes?
- Live portable monitoring system for detector tests at accelerators! Every time we test a detector with a beam at an accelerator, it is crucial to keep track of the environment around the setup. Let's build a device to monitor and record this information. In this way, it will be possible to disentangle effects due to the tested detector from those induced by changes in the neighbouring environment. Standard (temperature, pressure, humidity etc.) parameters should be monitored and recorded, along with more complex quantities (electronic noise, magnetic field variations, etc.). The device should be easy to transport and setup, complemented by a user friendly DAQ and also equipped with a series of alarms that can trigger on certain measured quantities. If all of this is accomplished, it may be possible to instrument an alignment survey feature?
- Resuspension of particles can be used to study contamination after nuclear fallouts on a microscopic level (more info can be found here). Dr. Kaminsky and Illinois GS Nico Santiago are researching this topic at Argonne. We can help them by carrying out studies on resuspension generated by pedestrians and vehicles!
    - Project 1: replicate the functionality of commercially available air quality monitors as well as expand upon them. One such sensor is the Purple Air PA-II which monitors concentration of dust particles in the air as well as temperature, pressure, and humidity. Additionally, we would like to have measurements of wind speed and direction. If time permits, this project would be made into a self-

contained, portable, weather-resistant, low-power, Internet of Things (IoT) device that transmits its data to a central server.
    - Project 2: Study pedestrian resuspension of particles on sidewalk surfaces. This project would look to create a data pool that does not currently exist for resuspension studies. A Plantower sensor would be attached to a follow pedestrian, walking a set distance behind the first pedestrian. The 2 should measure a constant distance between them, while the measurement is taking place. The purpose of this experiment is to study how many particles are kicked up during a walking cycle over a set distance by various types at different speeds.
- Are water fountains in Loomis equally powerful? If I am in a hurry, shall I choose a particular one? How do they compare to other buildings? Let's find it out with a PHYS371-built device!
- Smart jacket for cyclists: can we instrument that to display, on the biker's back, turning signals and brake lights? We can also try to track the biker's heart rate, distance traveled, etc. We can extrapolate these quantities into different measurements, such as total calories burned.
- Macroscopic Collider: The measurements of cross-sections are of great importance in particle physics. Can we design a setup to measure the (differential) cross-section of a macroscopic collision, for example, two air hockey pucks colliding with different impact parameters. What other physics phenomena can we measure with this apparatus?
- Add-on Roomba inertial tracking: where does it actually go when it's vacuuming the first floor of my house? Would a different navigational algorithm let it function more efficiently? How well can we integrate the rotations and accelerations of a Roomba autonomous vacuum cleaner to figure out where the device actually is?
- Spectral properties of African percussion instruments: Djembe vs. Conga. How (and why) does the sound change with technique?
- An initial feasibility study of a rotating-mirror arthroscope. Orthopedic surgeons use an optical instrument called an arthroscope to view the surgical field during procedures such as joint repair. The typical field of view of an arthroscope can vary from 75° to 115°. Could we expand this to greater than 200° by synchronizing image capture with the orientation of a rotating mirror?
- Instrumenting a sustainable paper-production laboratory. Professor Eric Benson (Art & Design) does research on the use of local fibers from corn, prairie grasses, etc.-- in the manufacture of paper. Variables such as temperature and humidity are probably important in various stages of the production process, but Benson's group isn't able to keep track of them while they work. We can fix that for them!
- Solar cell performance comparisons: control an NPN-based current source with an Arduino, see what various solar cells can do. I'm starting to use these in an agriculture technology project, and there are surprises in what I find. So let's scope this out in more detail.
- Predictive shock mitigation on Illinois Central passenger trains. Amtrak rails are a mess just south of Kankakee. Could the bumps felt in one car be radio'd to a device in a car further towards the rear? It might allow an active suspension supporting a crate of delicate devices to protect its cargo better.

- Noise produced by wind turbines. We want to do this in the time domain, not frequency domain.
- Multiple-head IR non-contact thermometer. What would it take to measure the temperatures of a dozen subjects simultaneously?
- Foot pressure profiles for users of standing desks. I like my standing desk, but should I be wearing protective footgear?
- Airborne particulate concentrations in agricultural settings (outside/inside tractor and/or combined cabins)
- Microphone-based, radio-linked vector anemometers (Prof. Gollin's invention!): could a sound engineer use these to correct (in real time) for wind-induced phase errors between towers of speakers in a large outdoor concert venue?

The style in which we will work

"DLP" stands for "Design Like a Physicist." That's a reasonably descriptive term for how we will go about things, though it wasn't my first choice for the three-character course identifier. Here is what I mean. If you took Physics 298owl from Prof. Gollin you'll remember that he had you hand-code many algorithms—integrators, Fourier transforms—that could also be found in professionally produced libraries. For pedagogical purposes, he had you reinventing many wheels.

That's not how we usually proceed in research or this course. If there's a pre-coded numerical algorithm we can use, we'll appropriate it, generally putting proper attribution to its source in comments in our own code. If there's a circuit we need that's described on an engineering website, we'll use it. Proper attribution can be placed on our schematic diagram. Sometimes it might be difficult to publish the source attribution—the 3D STL files you'll create for TinkerCad projects—but do keep in your own notes information about where you have found useful material.

You will keep track of your efforts in a physical paper (or digital) notebook in which you describe your work, useful revelations, and calculations. I do not want this to be anything fancy, but your notes should be cumulative rather than something you discard at the end of each class meeting. You should have your notebook open while working on your project. We won't be monitoring your compliance with this, but you really will find it useful. Be sure to date each of your entries. You should put notes about techniques you find (or invent) into your notebook so you can find them later.

You are to refrain from using your phones and laptops to visit social media sites, chat with friends, read news publications, and so forth. You are not to be looking at your tiny screens while any of us are presenting to the class. We (the instructor and/or TAs) will become very cross if we catch you doing this.

We will use several IDEs - Integrated Development Environments - during the semester. You are expected to install and use these in your work. If there are other tools that you'd prefer to use, keep in mind that it will be hard for you to share material with other members of your group. If you insist on staying with these, I will not be happy to find you wasting time translating your work into a mutually acceptable format.

You will be working with things for which your understanding will often be a little blurry. That's OK, and in fact that's the usual state of things in research. Taking the time to understand every last detail about an IDE is a waste of your time: it is better to focus your efforts on getting by, on muddling through. Mini-seminars about major tools useful for this class will be offered in the first half of the course, to give you a general idea of the tools and their capabilities, but they will far to be anything like a full software tutorial. You will get more done per week this way than you would if you spent the time to understand everything completely. There is too much to do and far more interesting things to consider than the arcane details of SPI and I2C interfaces. You want to understand them well enough to work with them, but not to write the definitive Handbuch der Was Auch Immer document without reference to external sources.

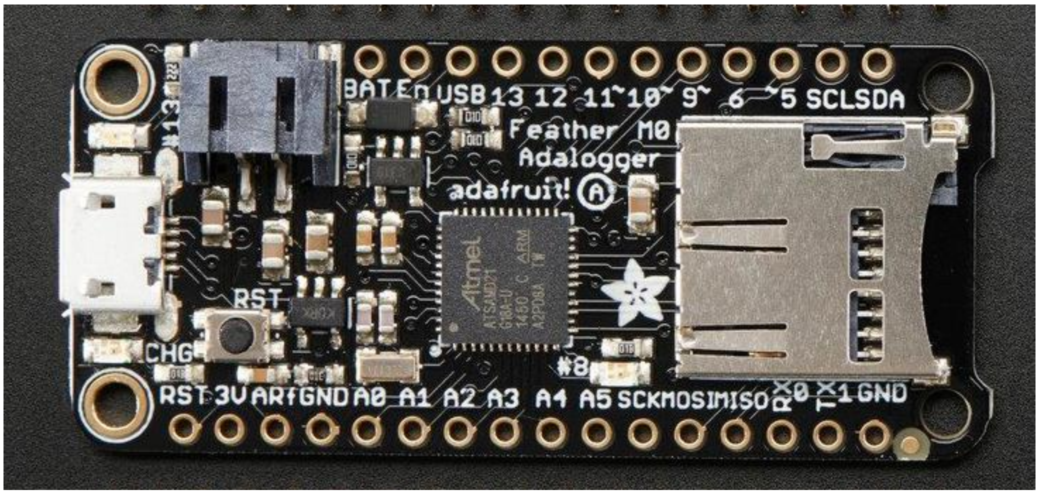
I will expect you to have these windows open on your laptop in class at all times: (1) the IDE for whatever you're doing; (2) a browser window with which you can search for (and download) useful things.

You will need to use a laptop that is able to read (micro)SD memory cards and to accept a standard USB-A connector. If your laptop can't do this, please acquire an adapter that will let you read a microSD memory card and another that will let you attach a USB-A cable, through which you will talk to the Arduino. I will have a few of these I can loan to you, but please get your own. Note that a "charger cable" WILL NOT work for you: it only carries ground and 5V lines, and omits the D+ and D- data lines. It's hard to tell the difference between a full-blown USB cable and a charger cable from external inspection. Figure below: USB-A on the left, USB-C on the right.

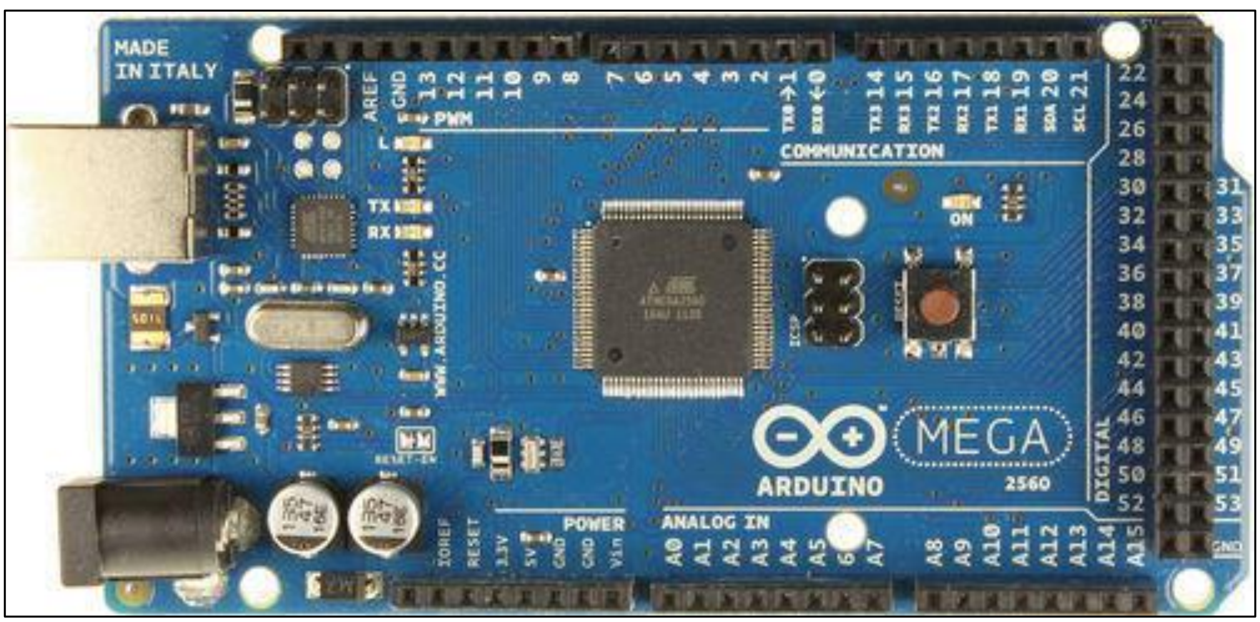
The tools we will use

Arduino coding IDE

The heart of your data logger will probably be either an Adalogger Feather M0 or an Arduino Mega 2560 microcontroller board. Pictures follow:



Adalogger Feather M0



Arduino Mega 2560

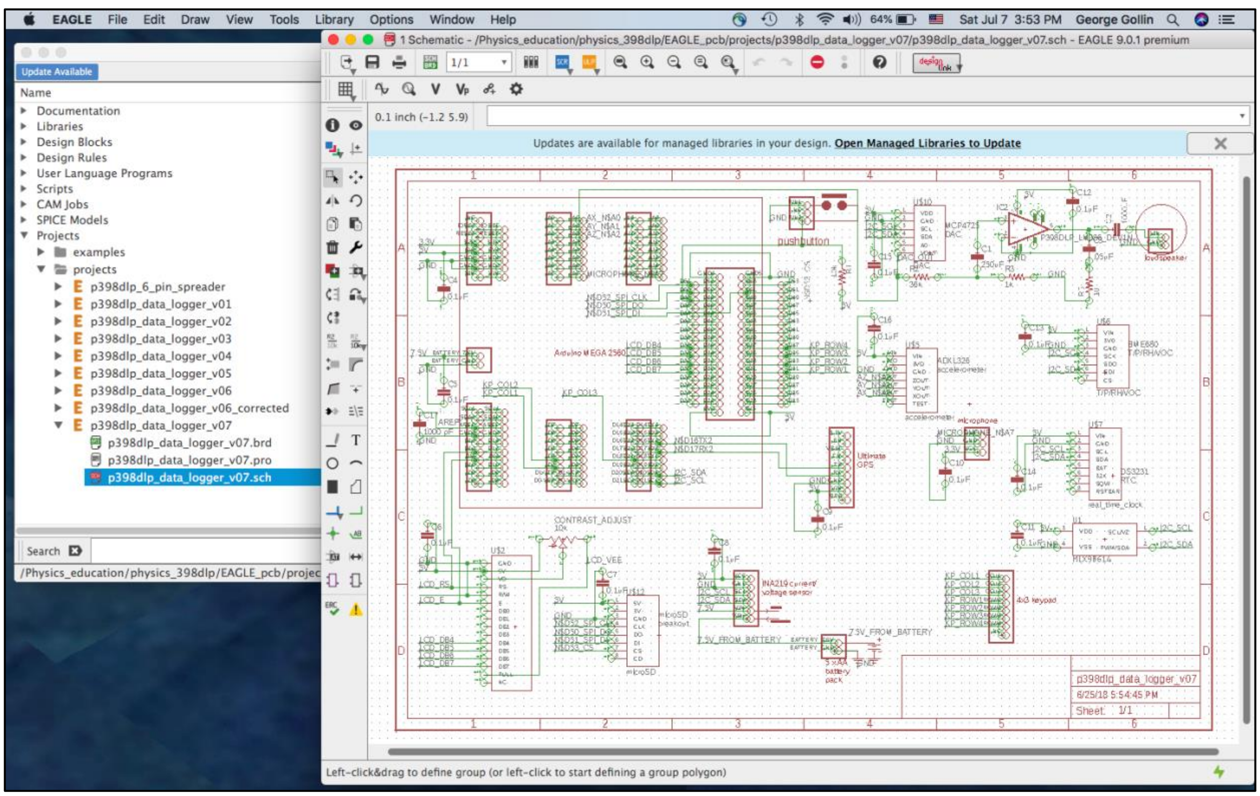
The Arduino is a remarkable little gizmo, featuring an Atmel Atmega2560 microcontroller running at 16MHz. The Atmega2560 has 256kB of flash memory in which your program will reside, along with 8kB of SRAM (static random access memory) in which the variables your program modifies as it executes will live. There are 16 analog inputs that feed an internal multiplexer whose output drives a successive approximation analog to digital converter.

The Arduino’s IDE (Integrated Development Environment) is quite a bit simpler than Anaconda’s iPython IDE. Most of what you will see on your screen is an editor window in which you will create/modify C++ programs that you will compile and upload to the Arduino.

Schematic capture

As you assemble your prototype on a breadboard, you’ll want to keep track of the wiring in your ever-more complex circuit. To do this, you’ll register an account with Autodesk and use their free-for-three-years EAGLE schematic capture tool. You will start with my version of the schematic and throw away the parts that you don’t plan to use.

Here’s a screenshot of the schematic capture tool. It might take you an hour to become reasonably proficient with it.



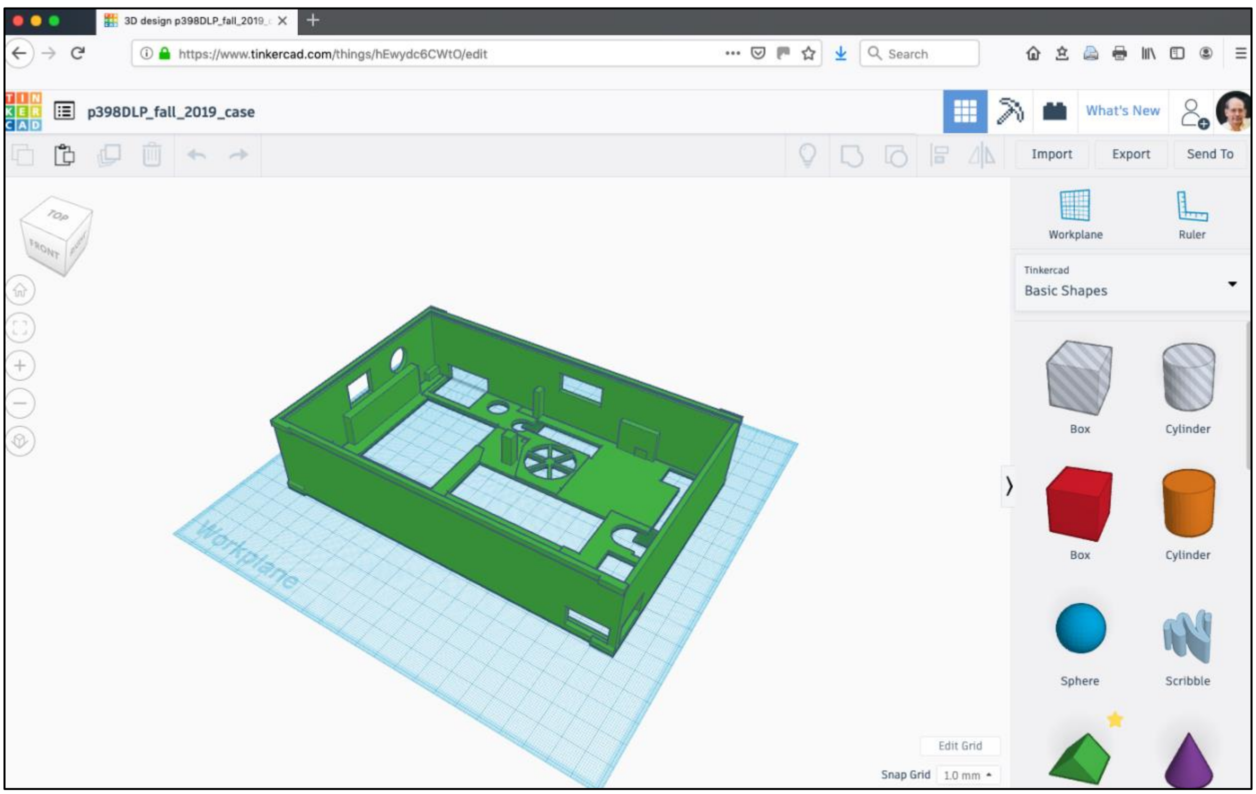
EAGLE schematic capture tool

We will have to design a PCB adequate for your application after you complete the assembly on your breadboard. Please take into account that a PCB takes typically 8-10 days to be delivered after the order is submitted.

3D printing
You will use TinkerCad—another Autodesk product—to design a case for your device. TinkerCad will produce an STL (stereolithography) file that we will feed to Cura, also an Autodesk application, to convert the STL file into a gcode file of instructions to be executed by a 3D printer.

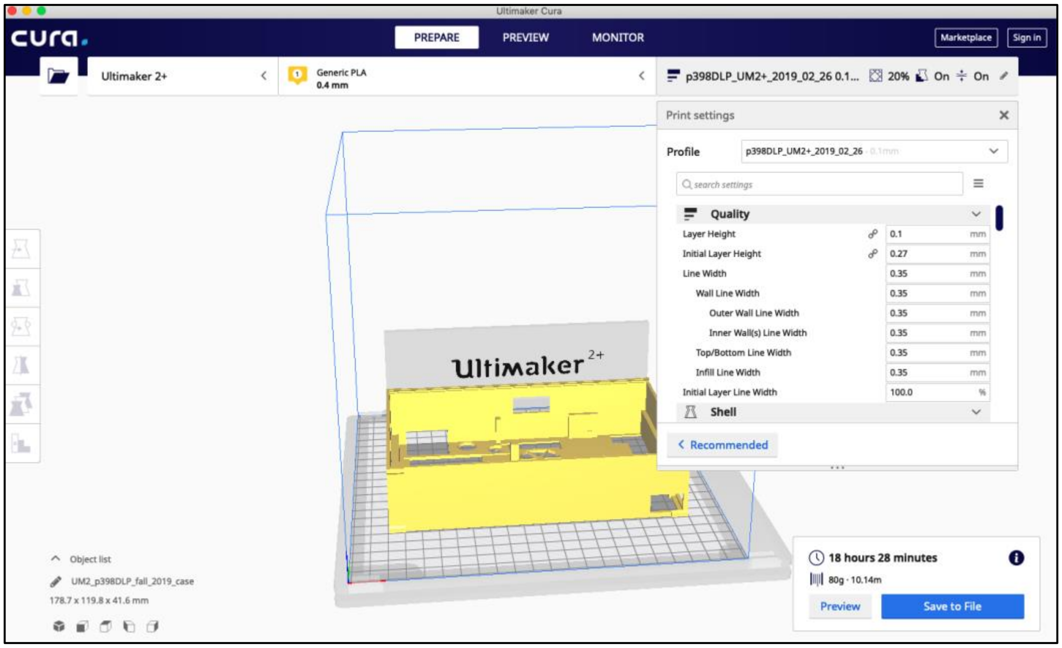
For practice you might consider designing a small box to hold a few of your sensor "breakout boards"; We can print these for you on the Ultimaker 2+ printers in Prof.Gollin's lab or using the the Creality3D CR-10S Pro 3D printer available in the ATLAS ZDC lab. When it comes time to fabricate the case for your data logger, I'll run the print jobs for you rather than negotiate with the Business School's MakerLab management.

Here's a TinkerCad screenshot of Prof.Gollin's Fall 2018 case design. TinkerCad is a web-based tool, easy to learn, and great fun to use. I have created the 2023 Class on TinkerCad, please register on it using this link: https://www.tinkercad.com/joinclass/VP9YD3TIK and your NetID as a nickname.



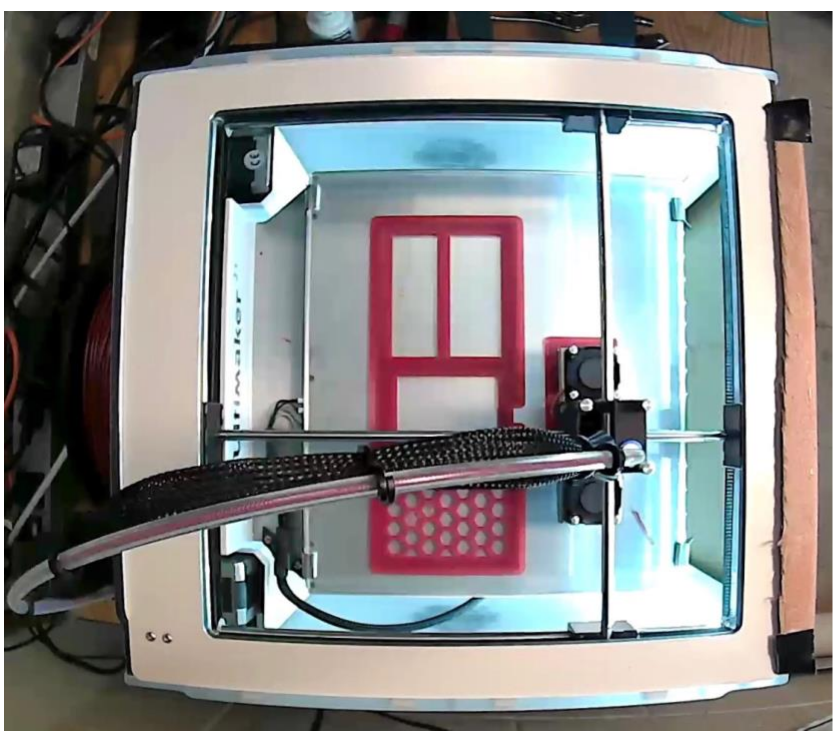
TinkerCad screen shot

Here's the Cura window. The August 2021 version (not the most recent!) is 4.10; you'll need to make sure you have the correct nozzle diameter (0.4 mm for the machines in my lab). You should also make sure that the “Preferences ➞ Printers ➞ Machine Settings ➞ Printer ➞ Origin at center” box is unchecked.



Cura screen shot

Here’s an Ultimaker 2+ printing the Fall 2018 version of the data logger case. The camera uses a fish eye lens, which is responsible for the distortion of the image.

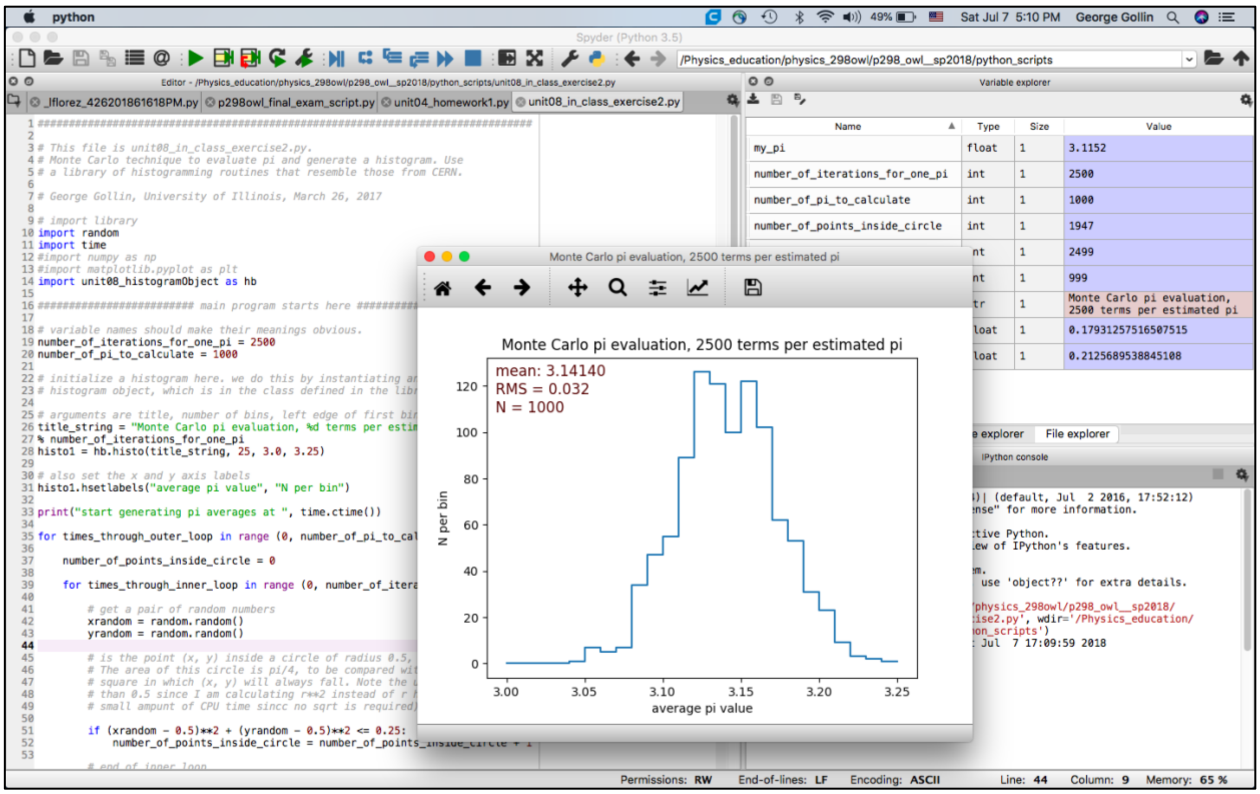


Ultimaker 2+ 3D printer

Offline data analysis

You should install Anaconda's Python spyder IDE for your data analysis work. Many of you are already familiar with Python, and with spyder. A copy of some of my Physics 298owl Python material from Prof.Gollin is available on the course webpage as an introduction/refresher to the language.

A screenshot of the Python IDE is below.

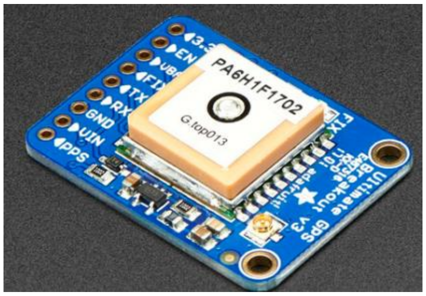


iPython IDE

Sensors and other hardware

Most of the sensors we'll use have been assembled onto small "breakout boards" by Adafruit Industries. I'll solder "pin headers" onto them so you can plug them into the breadboard you'll use when developing your prototype.

Here are photos of a few of the sensors, taken from the Adafruit web site. In order, left to right: GPS, amplified microphone, temperature-pressure-humidity-atmospheric volatile organic compound level. They are tiny, typically less than a square inch in size.



I have a partial list of sensors and other goodies that we have on hand in the Week 1 supplemental material document. Most of them are from Adafruit. I have enough of most of them for all of your breadboard and PCB devices, though depending on which project you do, you might use an Adalogger instead of Arduino processor.

If you'd like to fly an instrument package over, say, a farm, you can use the Physics 371 DJI Mavic 2 Pro drone. Note that you'll need to get a drone pilot's license before you can fly it.

DJI Mavic 2 Pro drone

Tools/materials we'll loan to each of you:

- wire stripper
- needle-nosed pliers
- tweezers
- eye protection
- breadboard
- digital multimeter
- spools of brightly colored 22 gauge solid-core wire
- Arduino Mega 2560

- USB-A to USB-B cord
- lots of breakout boards

So it's a pretty good collection of stuff. You are supposed to return all the material at the end of the course, in the last lecture. Your final grade will be validated only when the material is returned.

Learning objectives

You will learn to identify a measurement that you could make, in hopes of understanding more about, and perhaps improving, the State of Things.

You will learn to construct and build a device that might allow you to make those measurements.

You will learn to test and calibrate your device so that your data bear an understandable relationship to the physical parameters you are studying.

You will learn to do fieldwork, so that you come back from the world with valid data that can be analyzed and interpreted.

You will learn to report your results and speak of them to an audience of skeptics, supporting your conclusions with irrefutable facts.

You will also have a blast building your gizmo.

The rhythm of things

Time-on-task is an important part of mastering the tools you will use this semester. Rather than staging the various tasks for completing your project sequentially, you'll work with many of them in parallel. This will give you more time to digest the fine points of working with the tools we'll employ.

I expect that you will work closely (in an intellectual sense) with other members of your group, both in and out of class. You should spend at least six hours each week, outside of class, working on your project.

Speaking loosely, these are the things you'll do:

- select a project in collaboration with the other members of your group
- discuss measurements (and sensors) necessary for your project
- register a student, 3-years-free account with AutoCAD

- create a group code repository on GitLab
- install IDEs for Arduino C++, Anaconda Python, EAGLE schematic capture, and Cura
- wire up a breadboard version of your data logger
- write (or adapt) code to talk to all the sensors you are using
- write a data acquisition program
- update a schematic diagram so that it represents your device
- figure out how to take data and verify its validity
- load and test a PCB
- 3D-print a case
- do your field work
- write code to analyze your data and (cross) calibrate your sensors
- report your results.

The computer-based tool set comprises the following:

- Arduino programing IDE
- Anaconda iPython programming IDE.
- EAGLE schematic capture
- TinkerCad 3D printing design tool
- GitLab code versioning tool
- Cura 3D printing renderer, though I will probably run this for you.

Each week you'll advance the design of your data logger, write Arduino code to communicate with your sensors, further develop your plans for field work (including the structure of data acquisition code), and address physical infrastructure matters like case construction and PCB fabrication.

After the first week we'll have brief reports to the class from one of the teams. Topics (which I will assign) might include how a particular sensor works, what an interrupt does, how the I2C data transfer protocol works, and so forth. A report should last at most ten minutes, be carried in at most ten PowerPoint (or Keynote) slides, be presented to the class by all the team members, and be suitable for upload to the course web site.

Every week (times to be scheduled), each team will meet with me and the Tas for 30 minutes to discuss progress, problems, clever ideas, and any other issues that might arise.

Homework, exams, grading, milestones, and your obligations

The homework will consist of moving your design forward as far (and as fast) as you can. I expect you will spend about six hours at this outside of class every week. You should work with the other members of your team as much as possible, sharing code and design tips as convenient. You should document your progress, your plans, your brilliant realizations, your frustrations, and your concerns in your notebook.

You and your team will meet with me once per week in Loomis (location to be agreed upon) for 30 minutes. All members of your team must attend, and must arrive promptly at the scheduled time, without exception. You must always have on hand your laptop, breadboard circuit, notebook, and (once it is under construction) PCB circuit.

You and your team members will give several reports to the class over the course of the semester. The reports are to be clearly written PowerPoint (or Keynote) presentations (with proper attribution of sources) aimed at your audience of fellow students, who will not necessarily know what you mean by, for example, "I2C interface."

You must come to class on time, arriving with your laptop and power adapter, goodie box of parts and tools, and breadboard. When you have a PCB version of your device you should bring both the breadboarded and PCB versions of your device.

There will be no midterm exams. But there are milestones that I ask you to meet, and will consider these when evaluating your work for a grade. Here are the milestones and deadlines:

- 1a. Modify the Arduino's blink program so that it blinks the initials (of your English/American name) in Morse code. (Week 1, by end of Friday class)
- 1b. Install and test a BME680. (Week 1, by end of Friday class)
- 1c. On your breadboard, install the following devices (in addition to the BME680 and Arduino): LCD (including 10kΩ trimpot), keypad, and microSD breakout. (Week 2, by end of Friday class)
- 1d. Formulate a project plan and division of project responsibilities. (Week 2, by midweek group conference with course staff)
- 2a. Install, set, and read back a DS3231 real time clock. (Week 2, by end of Friday class)
- 2b. Install and read back a GPS module. Use it to set the DS3231 real time clock. (Week 2, by end of Friday class)
- 2c. Write a short text file to your SD card. Copy the file to your laptop, then write a short Python program to read it and display its contents. (Week 2, by end of Friday class)
- 2d. Finish installing all the parts on your breadboard required for your project's data logger. (Week 3, by beginning of Friday class)
- 2e. Register an Autodesk user account, then visit the TinkerCad website. (Week 3, by beginning of Friday class)

3a. Write a single bare-bones program that read all your project circuit's sensors and writes data to a microSD file. (Week 3, by end of Friday class)
3b. Write a single bare-bones Python data analysis program that generates histograms and plots of environmental data read by your BME680. Calculate means and RMS widths for these quantities. (Week 3, by end of Friday class)
3c. Log in to Autodesk and download EAGLE. (Week 4, by midweek group conference with course staff)
4a. Finish writing a reasonably sophisticated DAQ and use it for a quick field test of your devices. (Week 4, by end of Friday class)
4b. Analyze your field test data, generating the plots and calculations that you expect to appear in your ultimate report. (Week 4, by end of Friday class)
4c. Install breakout boards on your PCB and test it. (Week 5, by midweek group conference with course staff)
5a. Perform a longer set of field tests and run them through your analysis. (Week 5, by beginning of Friday class)
5b. In consultation with course staff, refine your offline analysis. (Week 5, by end of Friday class)
5c. Finish PCB and transition to using it for more field test data; verify that PCBs function as expected. (Week 5, by end of Friday class)
5d. Use TinkerCad to design personalized covers for your PCB cases. (Week 5, by end of Friday class)
6a. Take all the data that you think you'll need for your project. (Week 6, by end of Friday class)
6b. Verify that your data are valid: analyze them. (Week 6, by end of Friday class)
7a. Analyze production data and discuss your conclusions with course staff. (Week 7, by end of Friday class)
7b. Draft a modified run plan if appropriate, take more production data. (Week 8, by midweek group conference with course staff)
8a. Develop a detailed data analysis including cross calibration techniques, and run all your data through it. (Week 8, by end of Friday class)
8b. Write brief outline of a possible project report, discuss with course staff. (Week 8, by end of Friday class)
9-10. Write and submit "nearly final" draft of project report. (Week 10, by start of Friday class)
11-12. Rewrite and submit "final" project report. (Week 12, by start of Friday class)
13-14a. Prepare PowerPoint project presentation. (Week 14, by start of Friday class)
13-14b. Prepare and submit final project report. (Week 14, by start of Friday class)

In place of a final exam, I will require you and members of your team to generate two documents, with all team members as coauthors:

- A 30-minute PowerPoint (or Keynote) presentation describing your project: the measurement(s) you've made, and the reasons for so doing; the hardware and software you've built to perform these measurements; your fieldwork and calibrations; your analysis and conclusions. Your intended audience comprises your Physics 371 classmates. Presentations are due at the start of class, week 14; you will present during the 14th week of the term.
- A written report of approximately 20 pages (single spaced), describing your measurements and conclusions; this is essentially the same information that you will write into your PowerPoint (or Keynote) presentation. However, your report should be aimed at an audience that is outside the university community. For example, if you've measured the anomalous transverse accelerations of Amtrak trains, your audience might be the Illinois Department of Transportation. Your report is due in stages; see the course calendar for details.

About grading: if you work hard, are clever, come to all class activities, and do a good job on your hardware, software, and reports, you will receive at least an A. But if you miss class, miss conferences with me, are late uploading your diary files, miss milestones, or do not do a good job on your project or reports, I will hammer you.

If you miss class for a legitimate reason, you must submit documentation to Kate Shunk in the Undergraduate Physics Office or upload it through the excused absence portal linked to the "Course policies" page on the p371 website. Take note of the deadlines for submission of your documentation! I will expect you to make up for the work you didn't accomplish.

Your obligations include working in a safe manner: ***always*** wearing eye protection when soldering and ***always*** washing your hands soon after handling metallic objects such as header pins or solder.

Course components used to calculate your course grade:

- Short in-class group presentations on technical issues pertaining to your project;
- Quality and efficiency of your group's collaborative interaction;
- Quality of your project hardware and data acquisition work;
- Quality of your offline analysis software work;
- Compliance with schedule milestones and deadlines;
- Project PowerPoint (or Keynote) presentation;
- Project report.

Your grade will be based almost entirely on the semester's work, including your electronic diaries, and the final project reports and associated material.

Course staff

During the Spring 2023 semester the staff comprises one professor and one graduate teaching assistant. Depending on enrollment, the number of TAs might need to be increased.

The initial offerings of the course were taught by Professor George Gollin. However, any of the Physics faculty with significant experience in experimental techniques, including instrumentation, embedded systems, and electronics would be able to teach the course.

Office hours: On demand, via Zoom, by appointment. Each team will also have a 30-minute face-to-face meeting with course staff every week so we can monitor your progress.

Course calendar

what	day	date	comments
semester begins	Monday	January 17	
week 1	Friday	January 20	
week 2	Friday	January 27	see Celia Eliot's presentation on communicating clearly
week 3	Friday	February 3	
week 4	Friday	February 10	
week 5	Friday	February 17	
week 6	Friday	February 24	
week 7	Friday	March 3	
week 8	Friday	March 10	See Celia Eliot: “Writing clearly”
Spring Break 11-19 March			
week 9	Friday	March 24	
week 10	Friday	March 31	Nearly-final draft of report due
week 11	Friday	April 7	
week 12	Friday	April 14	Final draft of report due
week 13	Friday	April 21	
week 14	Friday	April 28	Rewritten report and presentation due
end of term conferences	Monday – Wednesday	May 1 – May 3	
reading day	Thursday	May 4	

Disability considerations

To obtain disability-related academic adjustments and/or auxiliary aids, students with disabilities must contact the course instructor and the Disability Resources and Educational Services (DRES) as soon as possible. To contact DRES, you may visit 1207 S. Oak St., Champaign, call

333-1970, e-mail disability@illinois.edu or go to the DRES website. If you are concerned you have a disability-related condition that is impacting your academic progress, you can talk with someone at the Counseling Center, McKinley Mental Health, or DRES about how to see a provider in order to obtain a diagnosis or get your questions answered.

Detailed syllabus; see the list a few pages back for milestones.

Week 1, in class: see the Week 1 supplemental material writeup.

- Form up into research groups of three or four people and begin discussing which project you'll pursue. Decide which sensors and other devices you'll use.
- Schedule a time for the team's weekly meeting with professor and TA.
- Install the Arduino programming IDE.
- Plug the Arduino into a USB port on your laptop, download the blinking LED sample program. Confirm that the LED really does blink. (You'll want to create a sensibly named folder to hold your Arduino programs.)
- Modify the blinking LED program to flash your first and last initials in Morse code.
- Duct-tape your Arduino to your breadboard and install a BME680, along with whatever libraries are necessary. Play with the BME680, taking note of its sensitivity to changes in atmospheric pressure.
- Install other components, as described in the Week 1 supplemental writeup.
- Install the latest version of Anaconda Python.
- Select/assign a group to report next week about the I2C communication protocol.

Post-class assignment (try to finish all of this before you meet with us next week)

- Formulate a plan of action with the members of your team. I want all of you to be involved with all flavors of activity: writing Arduino code, generating schematics to represent your devices, and so forth. But it is fine for one person to take the lead on, say, managing the code that interrogates the GPS package. Discuss with your group members your tentative plans for executing your project: who will focus on what; which sensors you will need; who you might need to contact for permission to enter their space for your measurements.
- Finish any unfinished tasks on the week's "in-class" list.
- Read the entire (printed) "Introduction and Syllabus"

Week 2, conference with the professor + TAs

- Discuss your project plan, including who you might contact for permission to, for example, install atmospheric methane detectors in the UIUC barns.
- Show us what your breadboard circuits can do.
- Describe sharing of project responsibilities among group members.

Week 2, in class

- Hear a group's "informative report."

- Install a DS3231 RTC and set its time.
- Install a GPS module and use it to set the RTC. Depending on where you are working, you may need to step outside when it's time to test your GPS board.
- Install everything you'll be using for your measurements onto the breadboard, but do not yet install any of the wiring required for any of the new breakout boards. Place a 0.1μF capacitor between +5V and ground close to each breakout board's power pins.
- One breakout board at a time, attach power and ground wires, and any other connections that are necessary to drive the board. (Note that you can "daisy chain" the I2C connections from board to board.) Find some demo software and make the Arduino talk to the board you've just wired. After it works, go on to the next breakout board.
- Select/assign a group to report next week about the BME680 T/P/RH sensors.

Post-class assignment

- Finish as many unfinished tasks on the week's "in-class" list as possible. (You should try to finish loading and wiring your breadboard, and talking to the individual sensors.)
- Register an account with Autodesk.
- Log in to TinkerCad using the link for the class given by the instructor, and make sure it recognizes your Autodesk account. Find a simple design for something amusing on the TinkerCad site and export it to an STL file after you place your initials on the object.

Week 3, conference with the professor + TAs

- Keep me apprised of your progress, and how you have decided to share the responsibilities for the various tasks needed to advance your project.
- Show us what you breadboards can do.

Week 3, in class

- Hear a group's "informative report."
- Informal (oral) progress reports from each group: describe your project to the class, and tell us your thoughts on how you'll approach it.
- Do initial DAQ code development, and start on offline Python work.
- Download EAGLE from the Autodesk site.
- Select/assign a group to report next week about how a successive approximation ADC works.

Post-class assignment

- Finish as many unfinished tasks on the week's "in-class" list as possible.
- Log into your Autodesk account, then install the EAGLE schematic capture/PCB IDE.
- Download and open the schematic for my version of the data logger, and delete everything except the components you will be using.
- Work on your DAQ code, and make sure it can write to the microSD card memory.

Week 4, conference with the professor + TAs

- Keep us apprised of your progress. The first version of your DAQ should be fairly far along by the time we meet.

Week 4, in class

- Hear a group's "informative report."
- Do a quick field test of all your breadboards.
- Analyze the field test data and make plots and histograms.
- Select/assign a group to report next week about electret microphones.
- Plus components into the PCB and begin testing it.
- Last ten minutes: whole class group discussion. How is it going? What's too hard/too easy? What are your thoughts about the field work you'll be doing? What kind of technical support might I provide to make your work go more smoothly? Have you made contact yet with anyone to learn the rules concerning entering into their environment to make measurements?

Post-class assignment

- Finish as many unfinished tasks on the week's "in-class" list as possible.
- Log into TinkerCad, find my sample case for the fall 2019 data logger, and copy it to your own account, then modify it to suit your tastes and preferences.
- Try to finish your DAQ.

Week 5, conference with the professor + TAs

- Progress reports from you, including how far you got with TinkerCad.
- Show us that your breadboard works, and show us how far along you've gotten your PCB.

Week 5, in class

- Hear a group's "informative report."
- Finish your DAQ.
- Finish PCB checking. It should be electrically equivalent to your breadboard.
- Field test the PCB; analyze the data.
- Refine your offline analysis.
- Use TinkerCad to finish any design work needed for your project.
- Select/assign a group to report next week about GPS.

Post-class assignment

- Run (and analyze) a longer field test.
- Generate plots of your test data; discuss possible modifications to your run plan based on what you see.
- Finish your TinkerCad work so I can generate a case for you.

Week 6, conference with the professor + TAs

- Progress reports from you, especially what you found in your field tests.
- Show us that your PCB works, and show us your case design.
- Tell us about your run plan and take production data.

Week 6, in class

- Five-minute (PowerPoint) progress reports from each group: describe your field tests.
- Hear a group's "informative report."
- Finish your DAQ and offline analysis code.
- Select/assign a group to report next week about GPS.

Post-class assignment

- Take data: all the data you think you'll need for your project.
- Generate analysis plots from your production run data.

Week 7, conference with the professor + TAs

- Progress reports from you, especially what you found in your data.
- Show us your offline analysis results and discuss possible conclusions to be drawn.
- Discuss your plans to determine calibrations for your sensors.

Week 7, in class

- Hear a group's "informative report."
- Refine offline analysis and consider modifying your run plan.
- Take more production data if appropriate.
- Select/assign a group to report next week about the Atmel ATmega2560 microcontroller.

Post-class assignment

- Perform whatever auxiliary measurements are necessary to calibrate your devices.
- Prepare a ten minute status report to present to the class about your field tests.

Week 8, conference with the professor + TAs

- Discuss your status report, calibrations, tentative conclusions, and online/offline code progress.
- Discuss possible modifications to your run plan, if it is warranted by what you found.

Week 8, in class

- Hear a group's "informative report."
- Ten minute status reports presented to the class, along with group discussions of calibrations, analysis techniques, and systematic uncertainties.
- Continue refining your offline analysis and conclusions.
- Begin outlining your project report.
- Select/assign a group to report next week about how an LSM9DS1 "9-axis" accelerometer works.

Post-class assignment

- Try to finish your analysis and draw your preliminary conclusions.
- Outline your project report paper. Your intended audience will be readers who are not familiar with Physics 371.

Week 9, conference with the professor + TAs

- Discuss your analysis, conclusions, and project report outline with us.

Week 9, in class

- Hear a group's "informative report."
- Write the first draft of your paper, discussing the details inside your group as you go: who writes which sections, and so forth.
- Further refine your offline analysis and conclusions if appropriate.
- Select/assign a group to report next week about "Pulse Width Modulation."

Post-class assignment

- Finish a substantial, nearly final draft of your project report.

Week 10, conference with the professor + TAs

- Give us a progress report on your paper.

Week 10, in class

- Hear a group's "informative report."
- Peer-reading/evaluating/commenting/correcting of your drafts.
- More work on paper and analysis as necessary.
- Select/assign a group to report next week about the MLX90614 infrared sensor.

Post-class assignment

- Write your "final draft" of the paper. You should think of this as the ultimate, polished, final version.

Week 11, conference with the professor + TAs

- Paper status

Week 11, in class

- Hear a group's "informative report."
- More analysis refinement and project report work.

Post-class assignment

- Finish writing the "final draft" of the paper.

Week 12, conference with the professor + TAs

- Paper status

Week 12, in class

- Peer-reading/evaluating/commenting/correcting of your project reports.

- Discuss in your group changes to be made to your paper.

Post-class assignment

- Begin rewriting your very final project report.

Week 13, conference with the professor + TAs

- We'll discuss corrections/improvements/rewrites you should make to your paper.

Week 13, in class

- Analysis tweaks and paper rewrites.
- Work on your project PowerPoint (or Keynote) presentation.

Post-class assignment

- Finish rewriting your project report.
- Finish your PowerPoint (or Keynote) presentation, ~30 minutes in length, describing your results to your classmates.

Week 14, conference with the professor + TAs

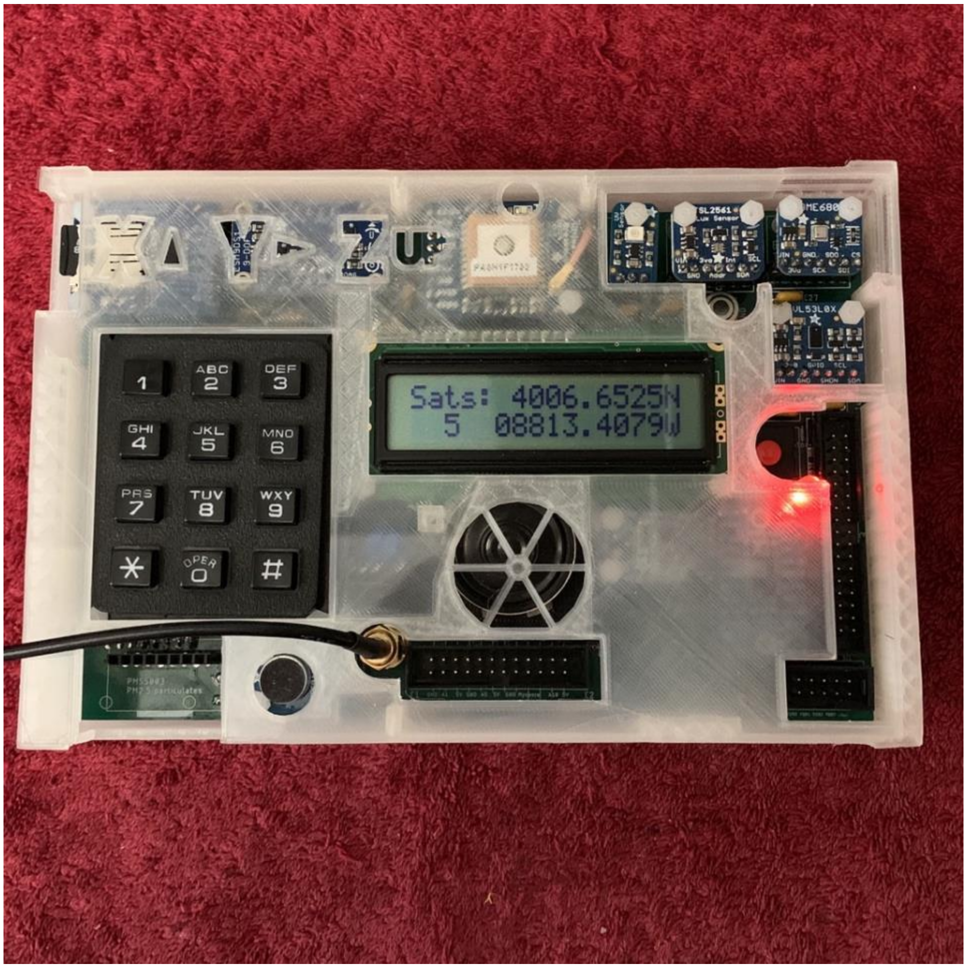
- Paper and PPT discussions.

Week 14, in class

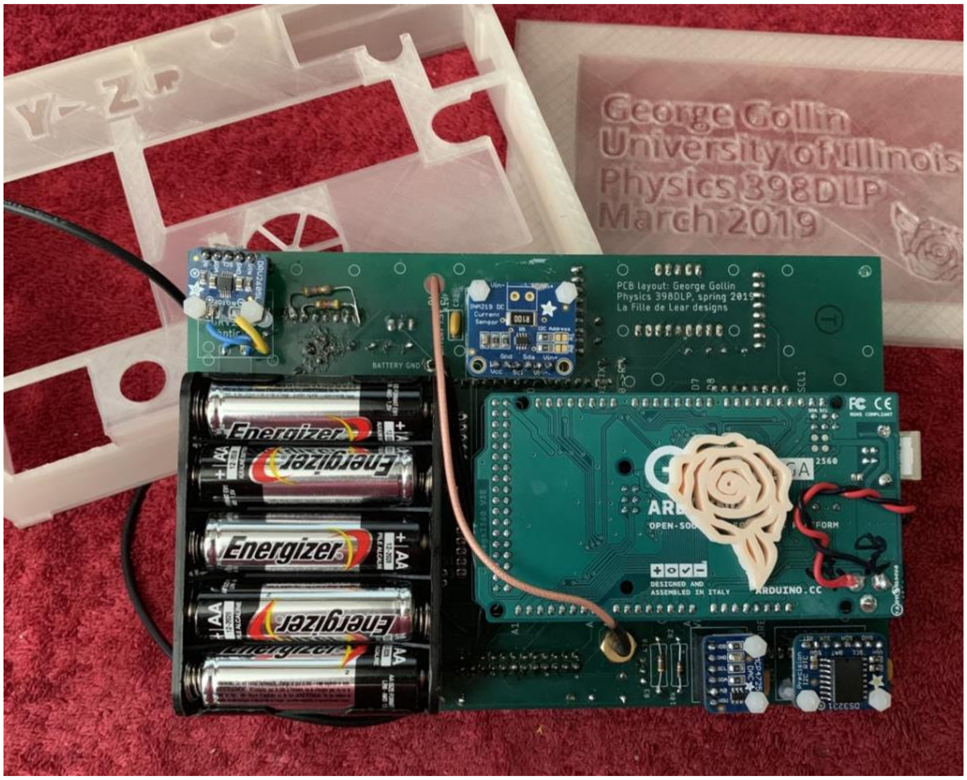
- PPT presentations by each group

A fully loaded data logger

Here are photos of my version of the data logger, onto which I've installed pretty much everything we have.


Sats: 4006.6525N
5 08813.4079W


George Gollin
University of Illinois
Physics 398DLP
March 2019
Sats: 4006.6525N
5 08813.4079W


George Gollin
University of Illinois
Physics 398DLP
March 2019
PCB layout: George Gollin
Physics 398DLP, spring 2019
La Fille de Lear designs
Energizer
ARDUINO.CC