Course name: Engineering 1

Version 1: Converting syllabus into UC_A2G process.

Basic Course Information

Title:	Engineering 1
Transcript abbreviations:	Engineering 1
Length of course:	Half Year
Subject area:	College-Preparatory Elective ("g") / Interdisciplinary
Integrated (Academics / CTE)?	No
Grade levels:	10th, 11th, 12th
UC honors designation?	No
Course learning environment:	Classroom Based

Course Description

Course overview (short):

Engineering 1 is a project based elective for students who want to create things by integrating hardware and software. We will be using the Arduino, a palm-sized programmable microcontroller, to build projects that interface with input sensors and output modules. There are 6 projects to build during the course, all open to extensions. The emphasis is on the technique and methodology, as well as on the engineering approach to creating a working project. This includes recognizing tradeoffs, and iterating on solutions to get to a viable result.

Course overview (long):

Engineering 1 is a project based elective for students who are interested in integrating hardware and software into their projects. We will be using the Arduino, a palm-sized programmable microcontroller, to interface with various input sensors and output devices to create embedded systems that accomplish a given task.

Students will be introduced to various aspects of engineering design and implementation, and will work on mini-projects throughout the semester. The mini-projects are designed to demonstrate the concepts learned, and to build student proficiency in designing with the Arduino platform. The students will build projects that require input devices (e.g. switches, light sensors, temperature sensors, distance sensors etc.) as well as output devices (e.g. motors, speakers, displays etc.). Some projects are very limited in scope and highlight specific subject, and will be done individually by the students. For example, using a temperature sensor and producing a visual output using light emitting diodes. Other projects are broader in scope, and

more open in implementation, and will be done in collaborative manner within a small group. For example, using light sensor to direct a robot to follow a flashlight.

The course culminates with a final project, which will allow the students to demonstrate their creativity and mastery of the techniques. Students can collaborate in small groups (up to 3) on the final project. The project will include researching a subject, identifying risks and difficulties in implementation, planning budget and timeline, and following iterative prototyping process to explore tradeoffs. Some projects implemented in the past are electronic harp, interactive art exhibit, robot to solve a maze, and air-pollution monitoring.
The work on the final project will start early in the semester, in parallel to the course material, to allow enough time for the process. Throughout the process, the students will communicate their progress to the class, and will work to create a digital portfolio that showcases their original proposal, research, prototyping process, obstacles faced, refinements made, lessons learned and a demo of their project. The course will end with a final presentation of the team projects to the broader school community.

Prerequisites: Introduction to programming (Python class)
Co-requisites: None

Course content:

The course is project based, and in each unit the students complete an original project that demonstrates their understanding of the subjects learned. In addition, each project includes reading and research components, as well as writing and communication elements. The reports are shared on the course web-page by the students, and at times are also shared as oral presentations.
Each unit description below includes the following 5 elements:

- Topics covered
- Project (as part of assignments)
- Research - Reading (as part of assignments)
- Report - Writing (as part of assignments)
- Formal assessment

From Unit 03, an additional component is added to the assignments: Final project. The students will have each week assignment related to the Final-project, helping them to stay on track.

Unit 01 – Introduction to Arduino
Duration: 1.5 week

Introduce the Arduino microcontroller and the development environment. (Resource: Primary textbook introduction).

Topics covered:

1. History of Arduino boards and micro-controllers
2. Major components on the Arduino board and different Arduino boards
3. Setting up the development environment: Hardware and Software
4. Running first program: Blinking light emitting diode (LED) on the board (Pin 13)

Assignments
a. Project: "Morse-code" – Modifying the Arduino program to transmit a pre-determined Morse code using the blinking LED.
Learning goals: Modify code; Verify (compile), download, and run phases.

b. Research: Morse-code – History, usage, automation and variations.
Learning goals: Researching resources, critical reading, and summarizing important aspects.

c. Report: Each of the projects will be described in a report. The goal of the report is to communicate the project to a general interested audience. The report will include (brief) verbal description, research as needed, images/videos, schematics (hand-drawn are ok), parts required, block diagram explaining the code, interesting specific issues concerning the hardware or software for the project. The report may include possible extensions. The report will also include the research subject if given. All reports will be published (by the students) on the course website.
Learning goals: Communication, reflection, extensions of a project.

* Formal assessment: Arduino board components. Microcontroller.

Unit 02 – Circuits
Duration: 2 weeks

Introduction to Circuits: Ohm's Law. Kirchhoff's rules. Introduction to multiple resistor circuits, switches, light emitting diodes, and named circuits. (Resource: Primary textbook chapter 1, and teacher notes)

Topics covered:
1. Ohm's law: Voltage source, resistors, diodes, switches
2. Kirchhoff's rules
3. Voltage divider, series and parallel resistors
4. Using multimeter to measure voltage, current, and resistance
5. How breadboard works
6. How to choose resistor to current-limit a light emitting diode
7. Connecting a circuit to Arduino
8. Connecting multiple elements in parallel/series

Assignments
a. Project: "Array of light emitting diodes" – A code controlled chain (or matrix) of light emitting diodes, and turning these on and off in a predefined sequence.

Learning goals: Design a schematic of a circuit, implement schematic, coding. Measure various quantities (Voltage, Current) from the circuit.

b. Research: Brief history of either Ohm or Kirchhoff (both historical figures).
Learning goals: See Unit 01.

c. Report: See details on reports in Unit 01. In this report, the students will need to add measurements carried on the project that demonstrate their understanding of Ohm's law and Kirchhoff's rules.
Learning goals: See Unit 01.

* Formal assessment: Simple electrical circuits. Analysis and design.

Unit 03 – Digital and Analog, potentiometer
Duration: 2 weeks

Introduction to digital and analog circuits. Pulse width modulation (PWM) to simulate analog output. (Resource: Primary textbook chapters 2 and 3)

Topics covered:

1. Digital and analog circuits
2. Digital logic: And, Or, Not gates and combinations
3. Potentiometer
4. Color (red, green, blue) light emitting diode, and controlling using pulse width modulation
5. Programming: Conditional statements, arrays

6. Introduction of Final-Project

Assignments
a. Project: "Perceiving color using only red, green, and blue diodes" – Write a program that produces a predefined sequence of colors.
Learning goals: Pulse Width Modulation (PWM), schematics.

b. Research: Color cube, and the spectrum emitted by each diode (read specifications). The student will need to explain how to create new color using discrete spectrums.
Learning goals: See Unit 01.

c. Report: See details on reports in Unit 01.
Learning goals: See Unit 01.

d. Final project: Brainstorm and research ideas. Students will work in small groups (of two or three students), and decide on plausible ideas for their final project. They should start a web-page for their project, and present in class (5 minutes) what the project is, why they want to do it, and what might be a risk(s) to completing it. In a second part of this assignment students will put a very rough schedule, with major mile-stones (and dates), and a list of special items (=budget) needed.

Learning goals: Research ideas, collaboration, planning (time, budget).

* Formal assessment: Digital and Analog. Logic using gates. Pulse width modulation.

Unit 04 – Inputs: buttons, photoresistor, and temperature sensor
Duration: 2.5 weeks
(Note: Extended by 0.5 week, as students will also need to work on their final-project during class time to get feedback and help from teacher and peers.)

Introduction to inputs. Pull-up configuration for reading an input pin. (Resource: Primary textbook chapters 5, 6 and 7)

Topics covered:

1. Push button, de-bounce
2. Photoresistor – Physics of operation and circuit model
3. Temperature sensor – Physics of operation and circuit model
4. Programming: Program flow control, for() loops
5. Serial monitor for communicating with the board

6. Final-Project: Mile stone 1

Assignments
a. Project: "In-N-Out" – Combines input-and-output. The students will need to choose a project to demonstrate the concepts, and some of the possible projects are: Modify the Morse-code project to take the input from a push-button, and translate it to light emitting diode output; Use the color producing project to reflect input from a temperature sensor; Using light-sensing to detect line-breaking.
Learning goals: Creatively suggest (or select) a project, design a schematic of a circuit, implement it, and code it.

b. Research: Bimetal switch – describe principal of operation, and possible usages for temperature sensing.
Learning goals: See Unit 01.

c. Report: See details on reports in Unit 01.
Learning goals: See Unit 01.

d. Final project: Progress report – Students should reach their first mile-stone. They will need to update their project web-page, and present in class the status. This will be a 'every two weeks' process.
Learning goals: Engineering trade-offs in their project.

* Formal assessment: Switches, buttons, photoresistors, temperature measurement.

Unit 05 – Communication with Arduino: Liquid Crystal Display (LCD) and Processing
Duration: 2 weeks

Connecting the liquid crystal display (LCD) to Arduino. Communicating between Processing and Arduino (Resource: Primary textbook chapter 15. Processing.org website)

Topics covered:

1. Liquid crystal display (LCD) – Physics of operation and manufacturing. Integration in circuit
2. Processing – installation of the Processing computer language, and integration with Arduino

3. Final-Project: Mile stone 2

Assignments
a. Project: Add to any of the previous projects a connection to either a liquid crystal display, or to Processing. For example, connect the light line-breaker (photoresistor project) to Processing to produce a sound every time the line of light is broken.
Learning goals: Reflection on and improvement of previous project.

b. Research: Compare liquid crystal displays to other types of displays (e-ink, cathode-ray-tube (CRT), Organic light emitting diodes (OLED)). These displays are taking such a big part of our daily-lives, that the subject merits a thorough understanding by the students.
Learning goals: See Unit 01.

c. Report: See details on reports in Unit 01.
Learning goals: See Unit 01.

d. Final project: Progress report. The students should reach their second mile-stone. They will need to update their project web-page, and present in class the status.
Learning goals: Engineering trade-offs in their project.

* Formal assessment: Display types.

Unit 06 – Motors and Servos- get things moving
Duration: 2 weeks

Driving a direct-current (DC) motor and a servo (Resource: Primary textbook chapters 8, 11 and 12)

Topics covered:

1. Motors and servos – Physics of operation, and integration in circuits
2. H-bridge to drive motors

3. Final-Project: Mile stone 3

Assignments
a. Project: "Button controlled robot" – Build a robot, including at least one-motor, that can be controlled by buttons. More advanced robots can be controlled by light sources or the use of other sensors or other input methods (e.g, Processing).
Learning goals: Create a moving robot. Motion and control loops. Driving a high-power component.

b. Research: Two subjects for research.
b.1. Control loops of dynamic systems – What are these, and how are these applicable to moving robots?
b.2. Motors – Different types. The students will create a poster that describes 3 different types of motors.
Learning goals: Control loops and motors.

c. Report: See details on reports in Unit 01.
Learning goals: See Unit 01.

d. Final project: Progress report – The students should reach their third mile-stone. They will need to update their project web-page, and present in class the status.

Learning goal: Engineering trade-offs in their project.

* Formal assessment: See above, poster on motors.

Unit 07 – Final project focus
Duration: 3 weeks

The last 3 weeks will be dedicated to the final project: Making sure the project works, web-page is well updated to the finish, and presentation and poster are ready for the final review. (Resource: Teacher and peers)

Topics covered:

1. Finishing a working project
2. Improving iterations, or trade-offs to scale down
3. Presentations: Web-page, presentation, poster

Assignments
a. Preparing all the components for the final project conclusion: Working project, web-page, poster, and oral-presentation.

Learning goals: Presenting work; trade-offs; working with a hard dead-line (presentation day, when we will have external visitors).

* Formal assessment: Final project.

Course Materials

Text Book

Main: "SparkFun Inventor's Kit Guide", by Sparkfun. Available online: www.sparkfun.com/sikguide
Optional 1 (Hardware focused): Exploring Arduino: Tools and Techniques for Engineering Wizardry , 1st edition 2013, Wiley, by Jeremy Blum
Optional 2 (Software focused): Programming Arduino: Getting Started with Sketches, 2nd Edition 2016, McGraw-Hill Education , by Simon Monk

Websites

Arduino	www.arduino.cc
Processing	www.processing.org
Sparkfun	www.sparkfun.com

====