

Description:

Technology is rapidly evolving and is becoming more integrated into our everyday lives. It is used in many ways to work smarter, not harder. However, technology is helpful to a certain degree -- it is useful when it is utilized properly and optimally. Hence, the more we learn about technology, the more use we can get out of it. Our goal is to write Arduino code to program our SeaGlide. In this unit, we are going to begin learning about the basics servo motors, color mixing LEDs and buoyancy engines. This lesson is adapted from the [Servos, IR & RGB](#) lesson provided by the U.S. Navy's SeaGlide DropBox.

Students will be able to:

- Code a servo motor
- Code a buoyancy engine
- Test and debug code
- Wire a circuit board to run motors

Students will understand:

Students will understand why servo motors and buoyancy engines are important and how they can use them. They will understand how to code motors so that they can be used on an underwater glider. Students will also understand how to code their buoyancy engines to control their underwater glider and how to debug and troubleshoot their code. Students will also test their own underwater gliders and buoyancy engines.

Key Definitions & Concepts [1]:

- **Servos:** a small electromechanical device that uses motors and sensors to produce a controlled motion.
- **Debugging:** trying to first isolate a problem in order to then correct it.
- **Buoyancy Engine:** Mechanism used to control buoyancy of the underwater glider

Standards [Copied from: 2]:

3A.CS.03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.

3B.AP.11 Evaluate algorithms in terms of their efficiency, correctness, and clarity.

Background Information

Prior Knowledge:

- Prior Python and Arduino Lessons
- Buoyancy Lesson

<p>Science Practices [Copied from: 3]:</p> <ul style="list-style-type: none"> • Asking Questions and Defining Problems • Developing and Using Models • Using Mathematics and Computational Thinking 	<p>Core Ideas [Copied from: 4]:</p> <ul style="list-style-type: none"> • Defining and Delimiting Engineering Problems • Developing Possible Solutions • Optimizing the Design Solution 	<p>Cross Cutting Concepts [Copied from: 5]:</p> <ul style="list-style-type: none"> • Cause and Effect • Systems and System Models
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------

Possible Preconceptions/Misconceptions:

Students may not understand that servo motors are different from buoyancy engines. In this lesson, the two are clearly defined so students will understand the difference. Students may also not realize, at first, how important debugging is until they test their underwater glider. This is an important concept in all of computer programming. If the students are struggling to decode their SeaGlide program, then refer their memory to prior Python lessons to recall how they decoded their own generated codes.

Lesson Plan - 5E(+) Model

Engage: Lesson #1: Servos & Buoyancy Engines [1]

Students will individually complete a brief *Review Quiz* on concepts from the previous arduino unit. This half sheet allows students to recall important information from previous days that are applicable within this unit plan. It is important for students to be able to recall this information so they can apply these understandings throughout this module's completion. The instructor should allow for students to complete the *Review Quiz* half sheet individually for 5 minutes, then facilitate an open class discussion to review students' answers and address any misconceptions as necessary. This section should take 10 minutes to complete.

Engage: Lesson #2: Testing & Debugging [1]

Students will individually complete a brief *Review Quiz* on concepts from Lesson #1: Servos & Buoyancy Engines within this unit plan. This worksheet allows students to recall important information from previous days that is applicable within this unit plan. It is important for students to be able to recall this information so they can apply these understandings throughout this module's completion. The instructor should allow for students to complete the *Review Quiz* worksheet individually for 5 minutes, then facilitate an open class discussion to review students' answers and address any misconceptions as necessary. This section should take 10 minutes to complete.

Explore: Lesson #1: Servos & Buoyancy Engines [1]

Part I: Introduction

Students will complete in pairs the *Motors and Troubleshooting* half sheet. This allows students to think about what are motors used for and why troubleshooting is useful and necessary. This will help get students in a better mindset to learn more in depth about how to code a servo motor for their SeaGlide.

They will also answer questions that will have them recall information about what debugging and troubleshooting are from previous Python lessons which is a necessary skill for students to grasp to build a SeaGlide and complete this module. Students should work with a partner for 5 minutes to complete the *Motors and Troubleshooting* half sheet. Then, the instructor should facilitate an open class discussion to review students' responses and address any misconceptions as necessary. This section should take 10 minutes to complete.

Part II: *Benchmark Lesson: Servos and Buoyancy Engines*

This part of the lesson covers the sections *3.1 Servo Motors* and *3.3 Buoyancy Engine* from the Arduino Unit 3 packet, [Servos, IR & RGB](#), provided by SeaGlide. Section *3.2 IR & RGB* is skipped because, while interesting, it is irrelevant to students' active needs for coding and building a SeaGlide. Reference the *Teacher Notes* attachment for materials lists and other information for this unit plan. Reference the *Instructor's Guide* to see how the packet should be read aloud while including stopping points for questions, activities and explanations.

Through the *3.1 Servo Motors* portion of the lesson, students will learn the basics of servo motors and how to code them to run on an Arduino board for their SeaGlide. Students will understand the usefulness and functionality of a servo, the utilization of internal feedback, and the difference between position controlled servos and continuous rotation servos. Students will use stock code that is provided from the packet to run two different sets of code; each code is designed to show the students the difference between position controlled servos and continuous rotation servos.

Through the *3.3 Buoyancy Engine* portion of the lesson, students will learn about buoyancy engines. Students will be provided with the stock code that is programmed into their SeaGlide for competition day. Through this lesson, students will complete the build of their SeaGlide (by finishing the building and the programming of the buoyancy engine). By completing the *3.3 Buoyancy Engine* portion of the lesson, students should have a deep understanding of what the Arduino code is written to do, of the wiring necessary for proper functionality, and of the function of a buoyancy engine. This lesson helps to prepare students for testing and troubleshooting their code and their build during the next day's lesson.

Explore: Lesson #2: Testing & Debugging

Part III: *Investigation Lesson: Testing and Debugging*

This part of the lesson is designed to be student-driven and student-led. Reference the *Instructor's Guide* for helpful resources that are provided by the U.S. Navy's SeaGlide website. The purpose of this investigation is for students to test and to debug the build and the programming of their SeaGlide. Students will have to make adjustments to their SeaGlide and might have to troubleshoot the code while at the pool. This should be freeform as students are meant to modify and test their underwater gliders independently and to explore the applied principles.

NOTE: Students should already have their SeaGlide completely built prior to this lesson because they will be testing their underwater gliders in a pool.

NOTE: It is recommended to have extra magnets available for the students since this is the only way for them to adjust the buoyancy of their gliders. Also, the build instructions provided by the U.S. Navy does not include a tether. Hence, instructors need to ensure that each SeaGlide has a tether attached to it (so students can pull their SeaGlide out of the pool).

Explain:

Throughout the exploration, the students will engage in discussions that inquire their understanding and knowledge of the information at-hand. Teachers should also be informally asking students to explain all topics and relevant connections throughout the entirety of this lesson when circulating the classroom. The worksheets include questions where students must explain their reasoning and troubleshoot their codes. This gives the students an opportunity to verbalize their understandings. Students will also have to be able to explain and find solutions to problems that they have with their underwater glider and buoyancy engine during the testing portion of this unit plan.

Elaborate:

Engines and motors are highly applicable in everyday life. Students see them all of the time, like in a car, or when observing an airplane. The material in this lesson is also directly applicable to building the underwater glider since students are programming elements that will be used to control their own underwater gliders. Hence, this lesson is highly applicable to the students' real-world experiences.

Evaluate:

Students are evaluated both informally and formally throughout the entirety of this lesson. The informal evaluations occur during the open class discussions. When the teacher is circulating the classroom, he/she is able to check for surface level understanding and make sure that the class is all on the same page by listening to students' discussions and responses. The instructor can gauge students' progress through observing whether or not they are struggling to complete the activities. The formal assessments of this lesson are the culmination of all the worksheets and review quizzes as well as whether or not the code runs properly. It is strongly recommended to instruct the students to return all documents used during each lesson for grading.

Enrich:

This lesson can be extended into a robotics or a physics classroom. Servo motors are readily applicable in the movement of machines and robotics. Also, servo motors can be used in multiple ways to control movement and direction outside of an underwater setting, which is most frequent. Hence, this lesson can be differentiated into a robotics or physics laboratory that studies power or energy from machines and engines, and that focuses on studying machines' movements.

****All associated documents are attached below****

****Reference *Annotated Bibliography* on the very last page of this packet****

Lesson #1: Servos & Buoyancy Engines: Worksheets

Teacher Notes for Reference Day 1 [6]

Use the [SIK Guide](#) to access the following: (including pictures and diagrams)

- Downloading and accessing Arduino pages 4-9
- Materials list and inventory pages 12-13
- A Single Servo build guide pages 49-52

NOTE: This reference does not have extra material on buoyancy engines.

Instructor's Guide Day 1: [1]

3.1: Servos

Have the students read the paragraphs of information starting on page 21 aloud, and stop reading on page 22 at the first exercise. Have the student pairs build the circuit as shown in *Figure 30* on page 21. Then, have the students upload the preset code titled “Servo Sweep” as shown in *Figure 31* on page 22 and complete the exercise. The students should be taking notes, writing observations and answering the exercises in their scientific notebooks throughout this process.

Once the student groups have finished the exercise, have the students read through the rest of page 22 through page 24 (until the exercise) aloud. It is recommended to have students read the Arduino code “Servo Knob” as shown in *Figure 23* aloud and to explain what the code is doing before uploading and running the code to their Arduino boards. After these discussions, have the student pairs upload the given code and complete the exercise on page 24. The students should be taking notes, writing observations and answering the exercises in their scientific notebooks throughout this process.

NOTE: This lesson skips over section 3.2 *IR and RGB*.

3.3: Buoyancy Engine

Ask the students the following questions:

- Why is buoyancy important?
 - Answer goal: Buoyancy allows us to raise and lower an underwater glider.
- How can we control buoyancy?
 - Answer goal: Changing the weight.

Have students read the paragraphs of information aloud starting on page 28 and ending on page 29. It is recommended to have the students read the Arduino code “SeaGlideV1_0.ino” aloud and to explain what the code is doing before uploading and running to code to their Arduino boards. Once these discussions have concluded, instruct the student groups to wire their boards as shown in *Figure 36* on page 28, and upload the “SeaGlideV1_0.ino” code to their Arduino boards. The students should be taking notes, and writing observations throughout this process.

NOTE: the final Arduino code upload is the EXACT code provided for the SeaGlide competition day. It is critical that students understand this code deeply enough to make any necessary edits during the final day of this unit plan.

NOTE: The testing referred to on page 29 will be completed on the final day of this unit plan.

Name: _____ Date: _____

Review Quiz [1]

What is Serial Communication?

What are methods?

Name: _____ Date: _____

Review Quiz [1]

What is Serial Communication?

What are methods?

Name: _____ Date: _____

Motors and Troubleshooting

What is a motor? How can they be used?

How are motors used on a SeaGlide?

What is troubleshooting in computer coding? Why is this important?

Name: _____ Date: _____

Motors and Troubleshooting

What is a motor? How can they be used?

How are motors used on a SeaGlide?

What is troubleshooting in computer coding? Why is this important?

Lesson #2: Testing & Debugging: Worksheets

Instructor's Guide Day 2: [1]

Helpful resources from the U.S. Navy's SeaGlide website:

[Build Guide Buoyancy Engine](#)

This PDF includes the basics behind building a buoyancy engine and includes a brief explanation of how the buoyancy engine in a SeaGlide works. This may serve useful for students in troubleshooting arising problems related to their build.

[Buoyancy Engine Assembly](#)

This PDF includes instructions for connecting the motors, engines, circuit boards, etc. to the underwater glider. This PDF also includes information about troubleshooting, which may serve useful for students understanding and fixing problems related to their build.

[Flight Principles](#)

This PDF includes the principles of flight for a SeaGlide that utilizes rudder positioning and additional magnets for balance. This may serve useful for students in enhancing their design as necessary.

[Flight Checklist](#)

This PDF includes both a pre-flight and a post-flight checklist to ensure that the underwater glider is ready for flight and is powered down properly. It is recommended to provide these checklists to the students prior to arriving at the pool for troubleshooting.

Name: _____ Date: _____

Review Quiz [1]

What is a servo motor?

What is a buoyancy engine?

Why is buoyancy important?

How can we test a buoyancy engine?

How do we make changes to buoyancy?

Lesson #1: Servos & Buoyancy Engines: Answer Keys

Name: _____ ANSWER KEY _____ Date: _____

Review Quiz [1]

What is Serial Communication?

A digital means of understanding what the processor is thinking.

What are methods?

They allow for the code to be broken up into tasks.

Discussion Notes: Methods, in Arduino coding, function similarly to functions in Python coding. Recall that a function, in Python, is either built-in or user-defined. In general, a function is a group of statements that perform a helpful task. Hence, a user defined function is a group of statements created by the programmer that performs a specific task within their program. One of the main reasons for defining functions is that they help break out programs into smaller, more sensible chunks. Utilizing functions is a great way to keep a code organized and manageable by avoiding repetition and making the code reusable. Methods, in Arduino, are useful for the same reasoning. The syntax between Arduino methods and Python functions is the sole difference.

Name: _____ ANSWER KEY _____ Date: _____

Motors and Troubleshooting [1]

What is a motor? How can they be used?

Motors are used in machine and convert electrical energy into mechanical energy. They allow for movement and control in machines.

How are motors used on an underwater glider?

They allow for movement and control of the underwater glider when moving up, down, forward, backward, left, and right.

What is troubleshooting in computer coding? Why is this important?

In coding it is trying to figure out why something is happening or not happening and trying to fix it then trying to isolate a problem in order to then correct it. This is important because it is essential in having a code that runs properly.

Lesson #2: Testing & Debugging: Answer Keys

Name: _____ ANSWER KEY _____ Date: _____

Review Quiz [1]

What is a servo motor?

A small electromechanical device that uses motors and sensors to produce a controlled motion.

What is a buoyancy engine?

A mechanism used to control buoyancy of the underwater glider

Why is buoyancy important?

Buoyancy allows for how much the underwater glider is going to rise or fall within the water. It provides the up and down movement of the underwater glider.

How can we test a buoyancy engine?

Answers may vary, but they should be along the lines of the following: testing the code to see if it works to lower and raise the underwater glider in the water.

How do we make changes to buoyancy?

Add/subtract weight to change the density. This is done by including magnets on the exterior of the SeaGlide.

Annotated Bibliography

- [1] SeaGlide. (n.d.). Retrieved from <http://seaglide.org/>
This lesson plan was used for excerption within the Arduino Unit 3: Servos and Buoyancy Engines. This referenced is used in completion of the explore in this unit.
- [2] Standards Aligned System. (n.d.). Retrieved from <https://www.pdesas.org/>
This website was used in each lesson in the Arduino modules to select proper Pennsylvania State standards, which are based in Common Core, that each lesson is centered around.
- [3] Nsta. (n.d.). Access the Next Generation Science Standards by Topic. Retrieved from <https://ngss.nsta.org/AccessStandardsByTopic.aspx>
This website used in every lesson in the Arduino Unit to find that are applicable in each lesson.
- [4] Nsta. (n.d.). Disciplinary Core Ideas. Retrieved from <https://ngss.nsta.org/DisciplinaryCoreIdeasTop.aspx>
This website was used in each lesson in the Measurements and Data Analysis unit to select appropriate disciplinary core ideas set forth by the NSTA that are at the center of each lesson.
- [5] Nsta. (n.d.). Crosscutting Concepts. Retrieved from <https://ngss.nsta.org/CrosscuttingConceptsFull.aspx>
This website was used in each lesson in the Arduino module to select appropriate crosscutting concepts set forth by the NSTA that apply to each mathematics lesson.
- [6] SIK Guide. (n.d.). Retrieved from <https://cdn.sparkfun.com/>
This reference is used for teacher guidance. It gives them another resource to use with the lesson.