Duckiesky Teacher Curriculum

Flight has fascinated humans for millenia.

The aim of this course is to empower people to build robots. Students will build, program, and fly an autonomous drone. This book covers everything needed to program an autonomous robot, including safety, networking, state estimation, controls, and high-level planning. Although the book focuses on an autonomous drone, we will provide a broad overview of modern robotics, including some topics relating to autonomous ground vehicles and robotic arms.

We will use the Duckiedrone to introduce concepts related to safety, control, state estimation, networking and communications, and mapping. Each student will build and program their own small quadcopter. After taking this course, students will be able to:

- Explain the space of designs for robotic communications, safety, state estimation, and control.
- Apply that knowledge to construct programs for communications, safety, state estimation, and control.
- Build, program, and operate an autonomous robot drone.

Contents

Section A - Introduction 4

- *Subsection A.1 - Importance of Robotics*
- *Subsection A.2 - Interacting with our Curriculum*
- *Subsection A.3 - Drone Operation*

Section B - Electronics 37

- *Subsection B.1 - Circuitry*
- *Subsection B.2 - Soldering*

Section C - Build: 1 59

Section D - Computing and Networking..60

- *Subsection D.1 - Using the Pi*

Section E - Sensors, Actuators, and Control: 1 71

- *Subsection E.1 - Overview*

Section F - Build: 2 76

Section G - Sensors, Actuators, and Control: 2..77

- *Subsection G.1 - Sensing*
- *Subsection G.2 - Middleware: ROS*

Section H - Build: 3 97

Section I - Sensors, Actuators, and Control: 3 98

- *Subsection I.1 - Motors*
- *Subsection I.2 - IMU*

Section J - Build: 4 108

Section K - Closed Loop Control 109

- *Subsection K.1 - PID Controller*

Section L - Localization 126

- *Subsection L.1 - Camera*

Section M - Materials 133

- Unit M.0.3 - Teacher Template 134

Section N - Bibliography 136

SECTION A
Introduction

The goal of this section is to motivate the course, as well as equip students with the background knowledge needed to consider the implications of autonomous flight. Additionally, this section gives students the ability to edit the textbook so that they can aid in its continual improvement.

This section contains three subsections: the first addresses the relevance of the course, the second introduces two tools that students can use to edit the course documentation, and the third is an introduction to the physics of flight and the hardware that makes autonomous flight possible.

We hope that this section gets students excited and geared up to build their drones.

Subsection A.1
Importance of Robotics

This subsection is designed to introduce autonomous drones within the historical and societal context of robotics. Robotis didn't simply "exist," and the ethical issues that technology poses are still major concerns. It is important to build up to autonomous flight to illustrate the process of scientific discovery and the evolution of technology. The goal is to show students how technology advanced in the past to the point where it is now, so that they are more aware of how they can advance today's technology into tomorrow.

Unit A.1.1
Intro to the Course

Student version (unknown ref duckiesky_high_school_student/introduction-relevance-society)

warning next (1 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/introduction-relevance-soci-
ety'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

Knowledge and activity graph

Requires:

Hardware -

- Pencil and paper

Previous lesson - N/A

Results:

Knowledge -

- What robotics is and why it matters
- Scope of the course

1.1. Robotic Marvels In Industries and Communities

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

ISTE 1. a.: Articulate and set personal learning goals, develop strategies leveraging technology to achieve them and reflect on the learning process itself to improve learning outcomes.

2) Assessments and Evidence of Understanding

By the end of this lesson, students should be able to explain why robotic technology is important in our day and age.

3) AGENDA (Brief Summary of Activities)

5 - 10 min: DO NOW set up with discussion

10 min: Robotic Marvels In Industries and Communities Lesson

20 min: Live Drone Demo

10 min: Groupd Research Exercise

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Teacher Materials

Screen to showcase students' DO NOW

Classroom Set Up

DO NOW and expectations written

1.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

Recommended: 5 - 10 minutes

DO NOW: Have students get into groups and try to write on a board more modern uses of robots than the other groups.

Hook:

- Robotics have become a seemless part of everyday life in the 21st century. As a generation raised in a technological era, we are able to easily overlook the idea of robotics. This will be an introduction lesson for students on the importance of robots for the world we are accostumed to.

2) Main Lesson

Recommended: 30 minutes

✔ For navigating the book, teachers may open the site on the board to add visual aid to the book explanation Or students may open it themselves by using their designated laptops.

Navigating the Book/Resources will be as follows:

- Sections
- Units
- Subsections
- References and Resources

By the end of the course, students will have a completed **quadcopter drone** capable of flying with commands inputted into a terminal on their computers! With the content in the following lessons students will dive into: Safety Regulations, Hardaware, Software, and Additional Content.

☀ Go over the vocabulary of a *robot* and a *quadcopter* found in the student book.

1.3. -Live Demo of Drone Here-

- The best way for instructors to Demo their drone safely is recomended in an outdoors area. If this is not an option, a containment cage may be required. Crafting these cages may require netting and PVC pipes. For the safety of instrcutors and students, the drone should be 6 feet away when taking off.
- Teachers can share their screen on a laptop while preparing to fly. The Javascript page with in the pidrone package can show the students the drone's control panel alongside the IR sensor's readings and camera image.

☀ Cover the *Why is it essential to learn about this? How do robotics help our communities in society?* section in the student book.

1) Ending The Lesson

✔ Exercise: Group Research: Students research in groups one ongoing robotics field that interest them to share with the class. Here are some examples:

1. Medicine
2. Transportation
3. Military
4. Space

Useful Resources and References

1. Lexico.com Definition of Robot
2. Wikepidia What is a Robot
3. Wikepidia Info on Robotics
4. Sciencing.com Use of Robotics

UNIT A.1.2
Development of Robotics

Student version (unknown ref duckiesky_high_school_student/introduction-relevance-history)

previous **warning** next (2 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/introduction-relevance-histo-
ry'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

KNOWLEDGE AND ACTIVITY GRAPH

Requires:

Hardware -

- Pencil and paper
- Laptop for research

Previous Lesson - N/A

Result:

Knowledge -

- General understanding of the history of robotics
- Scope of industries where robotics are applied
- Overview of state of the art robotics and what is in production

2.1. Robotic History and Evolution

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

ISTE 1. d.: Understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

2) Assessments and Evidence of Understanding

By the end of this lesson, students should understand the general development of robotics and be familiar with some of the emerging technologies in the field.

3) AGENDA (Brief Summary of Activities)

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Teacher Materials:

Could have visual aides/ slides prepared in advance of old technological inventions.

Classroom Set Up:

2.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

Recommended: 15 minutes

Hook:

- In this lesson we will take an in depth look at the evolution of robotics mechanics and inventions leading up to modern machines.

Note: Teachers can make references to the importance of robots in society from the previous lesson.

✔ Start off by showing this short fun clip on the evolution of Robotics (Wired Episode 1).

2) Main Lesson

Recommended: 20 minutes

Using the information in the student book and through the online resources below, have students do these optional activities:

✔ Exercise: 5-5-5: Have students get into groups, read a section from the student book (or a website) for 5 minutes, write notes bout the important points of it for 5 minutes, and present to the class about it for 5 minutes. This is a quick, interactive way to absorb information.

✔ Have students find images on their own of past technological achievements. Teachers could put various images of old inventions on the board and have students guess what their use is.

3) Ending The Lesson

Recommended: 10 minutes

✔ Exercise: Group research: Students can be tasked to research what robotic technology is being manufactured for the future to share with the class after reading the lesson.

Useful Resources and References

1. Standford University History of Robotics
2. Standford University Ancient Lore of Robots

3. Wikepidia Definition of Golems
4. Thought Co. History of Robotics
5. Wikepidia History of Robotics
6. Acieta.com Industry Use
7. Future Market Magazine Transport Robots
8. Automation.com Future of Robot Industries
9. Bliley.com Space Robots
10. Washington Post Space X Robot Program

UNIT A.1.3
Intro to Ethics

Student version (unknown ref duckiesky_high_school_student/introduction-relevance-ethics)

previous **warning** next (3 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/introduction-relevance-
ethics'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

KNOWLEDGE AND ACTIVITY GRAPH

Requires:

Hardware - N/A

Previous lesson - N/A

Results:

Knowledge -

- Students will be able to learn about what ethics is and why it is important
- Students will be introduced to ethical challenges in computer science and engineering including algorithmic bias, accessibility concerns, and human job security

Skills - N/A

3.1. Ethics

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

ISTE 2. a. cultivate and manage their digital identity and reputation and are aware of the permanence of their actions in the digital world.

Influence of Science, Engineering, and Technology on Society and the Natural World. Modern civilization depends on major technological systems. Engineers continuously modify these technological systems by applying scientific knowledge and engineering design practices to increase benefits while decreasing costs and risks.

Common Core ELA pg. 65: 2. b. Develop the topic with well-chosen, relevant, and sufficient facts, extended definitions, concrete details, quotations, or other information and examples appropriate to the audience's knowledge of the topic.

2) Assessments and Evidence of Understanding

By the end of this lesson, students will be able to understand ethical implications of technology and engineering. Students will be able to do presentations, debates, or other activities assigned to show their understanding and brainstorming of possible ethical consequences.

3) AGENDA (Brief Summary of Activities)

10 min: Introduction to the lesson. What is ethics and why it is important.

40 min: Introducing students to specific ethical implications of technological and engineering systems.

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Teacher Materials

- Projector for displaying videos or slides
- Large sheets of paper and markers for student brainstorming in small groups
- Print out note sheets for students to help take notes
- Reference the student textbook for more detailed explanations of concepts and examples

Classroom Set Up

- Space for students to be able to work and discuss in small groups.
- Make sure each group of students have at least one large sheet of paper and a number of markers.

3.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

> **Recommended:** 10 minutes

Hook

- There are many important decisions that need to be made for situations regularly. With drones or any AI or autonomous system that you are creating or using, students must understand that there are certain moral implications/ethics/principles that must be considered and have different consequences on others.

✔ Exercise: Group Brainstorm: Ask students brainstorm on paper in groups about ethics and important/common topics addressed by ethics. Have students come up with movie or pop culture references for ethical dilemmas surround technology and robotics.

- Examples of pop culture/movie references:
    - Spider-Man: Far From Home: Mysterio used autonomous and programmed drones to create the illusions, cause harm and chaos in society. Also Peter's ordered

drone strike on his classmate on the bus, example of almost civilian casualty/warfare.

- ∘ Avengers movies: Tony Stark's AI Jarvis, many instances of facial recognition in Tony's labs and suits, and the enemy Ultron (the whole let robots take over the world)
- ∘ Black Mirror TV show: many episodes revolving around technology and society. This article has particular examples of episodes and AI impacts on society

- Explain the origin of the word ethics and what is ethics:
  - ∘ The Universal Declaration of Human Rights includes a list of fundamental human rights which often closely relates to ethics
- Explain the importance of ethics and explain ethical implications:
  - ∘ what AI is : Bias and Fairness; Accountability and Remediability; Transparency, Interpretability, and Explainability
  - ∘ what AI does: Safety; Human-AI Interaction; Cybersecurity and Malicious Use; Privacy, Control, and Surveillance

2) Main Lesson

Recommended: 40 minutes

- In this lesson, students will learn about important ethical problems that are present in technology, autonomous systems, and engineering.

Correctness and Uncertainty of Algorithms and Autonomous Systems

✔ Exercise: Students brainstorm different autonomous systems they have heard about or used. Students should think about whether they trust these specific systems and why or why not?

- Explain autonomous systems, AI, ML, DL, and usage of algorithms.
- Explain the benefits and disadvantages of autonomous systems and the use of algorithms in decision making processes.

Example: Husky Vs Wolf in Image Identification:

- Explain an example of algorithmic inaccuracy due to unfair/unrepresentative data example with differentiating huskies and wolves.

☀ Paper about fairness involved in algorithms that undergo ML

Example: Artificial Neural Network Predicting Risk of Pneumonia Patients:

- Explain a simple algorithmic inaccuracy due to unfair/unrepresentative data example with identifying risk of pneumonia patients.

☀ Article with more detail about Pneumonia and Asthma Risk System

Example: The Boeing 737 MAX:

- Explain the incidents involving the Boeing 737 MAX.
- Explain why MCAS was added and how the MCAS and a faulty sensor contributed to the grounding of the Boeing 737 MAX.
- Explain other flaws with the incident.

Algorithmic Bias

✔ Exercise: Students should brainstorm about potential sources of algorithmic bias

✔ Exercise: Students can play this game to learn more about machine learning bias in the workforce.

- Explain that algorithmic bias can result from multiple sources.

Example: Congress Matched to Criminals by Rekognition:

- Explain another example of algorithmic inaccuracy, this one is regarding facial recognition which is more applicable to humans.
- Explain risks with its implementation in law enforcement.

An example: Autonomous Systems Identification by Skin Tone:

- Explain about a consistent problem in AI systems: how they identify people of different skin tones.
  - Autonomous soap dispenser and autonomous cars have trouble with this
- AI systems were less accurate at detecting people of darker skins by 5% (Georgia Institute of Tech).
- Identify possible reasons that contributed to these systems not working such as lack of diversity in workplace, IR sensors.

☀ article about encoded racial bias in technology

Another example: MIT's Moral Machine:

✔ Exercise: Students can try out some of the questions of the Moral Machine on this interactive website

- Explain that the Moral Machine focuses on nine different themes.
- Note about the correlation between results of the Moral Machine and culture and economics.

☀ Students may read this article to learn more about the experiment and findings.

> **Note:** While this is presented in a very game-like way, it is very interesting to take into consideration moral concepts that we think about in extreme situations.

Security/Systems Utilized in Society:

✔ Exercise: Students can brainstorm other ways that AI/autonomous systems/algorithms are used in systems used in society.

- Explain that there are many systems in society that utilize autonomous systems that are important to society.
- Explain open source and close sourced code.

✔ Exercise: Students think or write down about whether important systems/algorithms should be close or open sourced?

✔ Exercise: Based on what they have learned so far, has their original opinion changed about how much trust do we put into these autonomous systems?

An example: Use of Biometric Data in Society:

✔ Exercise: Students brainstorm about systems, apps, tools that use biometric data (ex:

FaceID, tracking down crime suspects, access to restricted buildings, access to important services such as healthcare)

- Explain advantages and disadvantages of biometric data in society:
  - Compare with pros and cons of other methods of verification (passwords, emails, hardware key)
- Importance of using multiple of these verification methods

☀ Article about Police Use of Facial Recognition

An example: Autonomous Systems Used in Social Credit System Development in China:

- Explain example of a system that utilizes AI: China's Social Credit System.

✔ Exercise: Class or group discussion about potential advantages and disadvantages of this system.

☀ article on China's Social Credit System

An example: Correctional Offender Management Profiling for Alternative Sanctions (COMPAS):

- Explain Correctional Offender Management Profiling for Alternative Sanctions (COMPAS).
- Explain that COMPAS has displayed bias against African Americans.

☀ Article

An example: Unsecured/Exposed Robots Running on ROS and Internet::

- Teachers can teach about risks of vulnerable systems can be when connected to the internet.

☀ Article 1 and Article 2

> **Note:** Students will learn more about Robotics Operating System (ROS) in a later module, but it is a set of open source libraries that can help with programming of robots.

Militarization

- Explain the use of autonomous systems for militarization (to make military decisions, or to take direct action).

International Traffic in Arms Regulations (ITAR):

- Explain International Traffic in Arms Regulations (ITAR) and what it covers.

☀ website

Example: Predator drones utilized by the United States:

- Explain predator drones as an example of drone systems that are in use by the US government for military operations.
- Unmanned aerial vehicles face many ethical issues such as civillian casualties.

✔ Exercise: Students can write a paper/debate in groups about whether they think predator drones should be used.

Autonomous Weapons and Chemical Weapons:

- Explain the different types of chemical warfare agents in US.

  ☀ History of US Chemical Weapons Elimination

  ☀ Class discussion: autonomous weapons should be treated like chemical weapons? Should they also be prohibited from use and fully destroyed?

Medical, Healthcare, and Caregiver Robots

✔ Exercise: Students can brainstorm where AI systems are utilized in healthcare.

- Explain that there are many benefits, but there are important ethical implications:
  - Privacy/security, trust between robots and humans, and their interactions

Example: The Emergency Exit Robot Study, Georgia Tech Howard:

✔ Exercise: Show of hands by students: would you follow a robot during an emergency?

✔ Exercise: Show of hands: What if it seems to be going in a not so great path?

- Explain about experiment conducted by Georgia Institute of Technology.

  ☀ Study

Availability/Accessibility/Uses

- Explain robots are available to the public vs private sectors, accessibility of robots to people who may have accessibility issues, and the large amount of uses of UAV and autonomous systems.
- Explain the effect of robot costs on consumers.
- Explain that robots can help with humanitarian and emergency efforts.

UN Guidelines for Emergency Uses of Drones:

- Explain different uses of drones and UAVs by the UN.

  ☀ Read this article for more information on UN and drones.

  ☀ Read this article about UN's operation with unmanned aircraft in DR Congo.

  ☀ Read this article to learn more about Drones and Humanitarian Action.

Future impact of AI on human jobs and responsibilities

- Explain that with the development of AI and technology, there has been a growing reliance on them as tools in our daily lives.
- Interesting video that can be watched:
- Explain ethical implications related to what AI can impact:
  - Automation, Job Loss, Labor Trends; Impact to Democracy and Civil Rights; Human-Human or Human-Agent interaction.

3) Ending The Lesson

Recommended: 5 minutes

- Summarize ethics and the large positive and negative benefits of AI.
- Remind students that there will be a safety module for the course coming up in one of the future classes.

✔ Exercise: Teachers could also assign a short report/presentation/essay to students on a specific section of this lesson or overall how their opinions have changed before and after the lesson.

Useful Resources and References

1. Seattle Times Article for more information about the MCAS system for the Boeing 737 MAX:
2. Verge Article about other flaws involved in the Boeing incidents
3. Washington Post Article on the lack of notice to FAA about Boeing MCAS system
4. Paper on Predictive Inequity in Object Detection
5. Moral Machine Test
6. Paper on the Moral Machine Experiment
7. Interactive moral machine
8. Article analyzing results from different countries
9. Paper about Fairness involved in Algorithms that undergo ML
10. Article with more detail about Pneumonia and Asthma Risk System and Wolf Vs Husky Identifier
11. Article about Rekognition and its failed Congress classifications

SUBSECTION A.2
Interacting with our Curriculum

The goal of this subsection is to introduce two tools that will allow students to edit the curriculum. Student involvement in modifying the course documentation is invaluable to its improvement.

The first tool is GitHub, which is a version control software; that is, it is like a folder with files, where each file has a record of all of the changes that have been made to it. The second tool is Markdown, which is the first programming language introduced in this course. Markdown is used to format text. Instead of using the toolbars Microsoft Word or Google Docs, Markdown allows writes to change the style of the text just by using symbols around the text.

The tools introduced in this subsection are very widely used. There are many tutorials on their use, as well. In some lessons, we link to tutorials that introduce the tools better than we could ourselves. If you discover any teaching insights on these topics, please leave a Git issue (taught in this section) to let us know, or add resources that you found useful for students.

Unit A.2.1
Git and Github

Student version (unknown ref duckiesky_high_school_student/introduction-curriculum-github)

previous **warning** next (4 of 36) index
warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/introduction-curriculum-
github'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

Knowledge and activity graph

Requires:

Hardware - Basestation

Previous lesson - N/A

Results:

Knowledge -

- Importance of version control in general
- What are local and remote git repositories and their purpose
- GitHub vocabulary (repository, pull request, fork, clone, commit, issue)
- Confidence in their ability to report back problems in or suggestions about the curriculum

Skills - Ability to use GitHub GUI to fork a repository and submit issues

1.1. Git and Github

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

ISTE: 1. c.: Use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways.

2) Assessments and Evidence of Understanding

By the end of this lesson, students should be able to navigate the GitHub GUI and recognize basic git commands.

3) AGENDA (Brief Summary of Activities)

5 min: Computer Setup

15 min: Git and GitHub Motivations

40 min: Git and GitHub Tutorials

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Teacher Materials:

Basestation, a projector (optional), worksheets printed out

Classroom Set Up:

Teachers can write a DO NOW on the board for students to set up their basestations and open a web browser and have them complete the creation of a GitHub account process (step 1 in the student book) while you introduce the lesson. Teachers can hand out the worksheets located in the student book at this point if they are printed.

1.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

> **Recommended:** 5 minutes

Hook:

- This will be a lesson for the students on understanding Git and navigating GitHub. It is important to learn how to utilize version control systems for most code-based projects. The DuckieSky textbooks are managed through GitHub, so the hope is that students will be able to directly edit and submit possible changes in our textbook to the DuckieSky Team, directly!
- (Optional) Conceptually connect GitHub to Google Drive in the sense that both platforms host file systems that multiple people can upload, download, and edit from. Emphasis that GitHub and Git are catered towards code-based projects.

2) Main Lesson

> **Recommended:** 15 minutes

Teachers should refer to the student book to either cover the following sections or direct students to read and interact with the following sections.

1. Students create a GitHub account if they do not already have one.
2. Students learn what Git and GitHub are used for.
3. Students learn why we are using GitHub.

3) Ending The Lesson

> **Recommended:** 40 minutes

- Students learn the basics of Git/GitHub.
  - The basics of GitHub can either be explained by the teacher or by having students learn through the tutorials in the student book at the teacher's discretion.
    a. If preferred for the Forking Tutorial's cloning and committing section, the teacher can demo the process.
  - Teachers should emphasize to the students that they are learning this material and/or going through these tutorials to be able to directly interact with our drone curriculum in a more comprehensive manner, which will become more clear next lecture.

Students need to know the following features of Git and GitHub by the end of this lesson:

1. How to create a GitHub repository
2. Creating and handling GitHub branches
3. Making and committing and changes to GitHub repositories
4. Forking a repository
5. Cloning a repository
6. Making and merging a pull request
7. Making a GitHub issue

✔ Exercise: Teachers can have students fill out the accompanying GitHub Defintions worksheet either while they are going through the tutorials or after they are completed with them. Here is the answer key if necessary.

Useful Resources and References

1. GitHub Guides
2. GitHub Glossary

Unit A.2.2
Markdown and Contributions

Student version (unknown ref duckiesky_high_school_student/introduction-curriculum-editing)

previous **warning** next (5 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/introduction-curriculum-edit-
ing'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

Knowledge and activity graph

Requires:

Hardware - Basestation

Previous lesson - Git and Github

Results:

Knowledge - Definition and uses of markdown

Skills -

- Ability to program in Markdown via formating and styling text using the Markdown language
- Ability to submit pull requests to the curriculum

2.1. Markdown and Contributions

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

ISTE: 1. c.: Use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways.

2) Assessments and Evidence of Understanding

By the end of this lesson, students should be able to submit a pull request to alter a Duckiesky page.

3) AGENDA (Brief Summary of Activities)

5 min: Basestation Setup

40 min: Duckiesky Documents Tutorial and Markdown/Markduck

15 min: Editing a DuckieSky Page

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Teacher Materials

Basestation, a projector (optional)

Classroom Set Up:

Teachers can write a DO NOW on the board for students to start up their computers/devices and open up a web browser.

2.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

> **Recommended:** 5 minutes

Hook:

- This will be a lesson for the student on editing and adding to the DuckieSky documents. It is important to learn how to propose changes and alter the documents that the students are learning from. the more helpful you are in helping us improve the materials for this drone course, the better the course experience will be for you and for students learning this curriculum in the future!

2) Main Lesson

> **Recommended:** 35 minutes

- Students will be able to submit pull requests to make changes to the books directly that can be approved by the DuckieSky team!
    - The book pages are written in Markduck.
- Students learn what Markdown and Marduck are used for
- Students learn basic Markdown/Markduck

> ☀ The basics of Mardown/Markduck can either be explained by the teacher or through the student book's instructions at the teacher's discretion.

Students should be aware of the following features in Markduck:

1. Wrapping text with two asterices or one underscore both at the beginning and at the end of the phrase **bolds** or *italicizes* respectively.
2. Adding a hash # with a space in front of text on a new line creates a heading; adding more hashes will reduce this heading size (e.g. ### Hello).
    - In Markduck, headers are used to define pages and separate sections.
3. Creating a hyperlink by wrapping the text in brackets [] and putting the link in parenthesis (e.g. Visit GitHub!)
4. Creating an image with similar syntax to links but adding an exclamation mark (!)

before the text.

- Adding a figure in Markduck

5. Creating an ordered list by putting subsequent numbers followed by a period on each new line starting with 1. with text after each number (e.g. **enter** 1. Cat **enter** 2. Dog **enter** 3. Hamster); Creating an unordered list by putting a dash (-) in Markduck instead of () *on each new line with text after each dash (e.g. enter - Cat enter - Dog enter** - Hamster).

- Additionally, sublists are created by using tabs instead of spaces in Markduck.

6. Adding two spaces to the end of a line of text and entering will create a new paragraph.
7. Knowing about the following special paragraph tags

3) Ending The Lesson

> **Recommended:** 20 minutes

- Teachers either can do a live demo of the pull request and issue process in the two videos linked in the student book, show the video to the class and speak through the steps, or let students watch the videos and read the instructions.
    - This is the location of the student-based book that can be used to demonstrate the repository.
- (Optional) Teachers demonstrate how our textbooks are hosted through Github and how individual pages in the textbook are written in Markduck.
    - Here is the link to the GitHub repo for the student book with the markdown files for the textbook pages (which can also be accessed by clicking the pencil in the textbook output). Teachers can show the output and the repo to compare how the structures are similar.
    - Teachers can either click through other pages in the textbook or otherwise encourage students to click through the files and folders in our textbook repo.

✓ Exercise: Students should submit a pull request to change the next page, **Students: Leave your mark here!**, with their newly acquired Markduck skills and GitHub skills from last lesson!

(Optional) Teachers can checkoff the pull request that students made.

Teachers should emphasize to students that if they see problems/areas that could be improved with the course, they can report and even fix them!

Useful Resources and References

1. Basic Markduck Guide
2. Local Duckiesky Editing and Docker
3. The Duckietown documentation on how to edit and submit pull requests the documentation with GitHub.

Unit A.2.3
Students: Leave your mark here!

Student version (unknown ref duckiesky_high_school_student/introduction-curriculum-student)

previous **warning** next (6 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/introduction-curriculum-stu-
dent'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

Refer to student book section to have students submit a pull request to change.

SUBSECTION A.3
Drone Operation

The goal of this subsection is to explain the anatomy of a robot: what hardware and software is essential to all robots. Then, this abstract anatomy is made concrete with the hardware and software that is used on the drone for autonomous flight. Finally, this section introduces the safety precautions that must be taken when working with flying robots.

Unit A.3.1
Sensors and Actuators

Student version (unknown ref duckiesky_high_school_student/introduction-operation-sensors)

previous **warning** next (7 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/introduction-operation-sen-
sors'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

Knowledge and activity graph

Requires:

Hardware - Unassembled drone kit

Previous lesson - N/A

Results:

Knowledge -

- Basic components of robot hardware and software
- Your specific drone’s sensors and actuators
- How the sensors and actuators interact to achieve stable flight

Skills -

- The ability to identify all of the sensors and actuators of your drone

1.1. Lesson 1: Anatomy of a Robot

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

NGSS: HS - PS4 - 5: Communicate technical information about how some technological devices use the principles of wave behavior and wave interactions with matter to transmit and capture information and energy.

ISTE: 1. d.: Understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

2) Assessments and Evidence of Understanding

By the end of this lesson, students should be able to describe the basic components of a robot and identify every part in their drone kit.

3) AGENDA (Brief Summary of Activities)

15 min: Discussion questions about robot anatomy

35 min: Learn drone parts with worksheet activity

10 min: Game to assess ability to identify drone parts

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Teacher Materials:

The drone kit, worksheets printed out

Classroom Set Up:

The students' drone kits can be placed at their desks before class

1.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

> **Recommended:** 15 minutes

Hook:

Before students start building their drone, they need to be able to identify all of the components in their drone kit and, on a surface level, understand how they interact with each other. Even though these parts are specific to our drone, every robot has some combination of sensors, actuators, and controllers to accomplish their goal. Emphasize that, by the end of this class, this pile of parts will be a working drone, and knowing what the parts are is the first step to accomplishing that.

1. Introduce Important Vocabulary:

Definitions found in the student book here.

1. Class Discussion Questions:

> Note: This is an optional discussion based on the engagement of your class. If this seems too basic, feel free to skip through the introduction.

Q: What sensors to humans have to percieve the world around them?

Answer: Eyes, ears, nose, taste buds, sense of touch, etc.

Q: What types of robots would need the same sensors you have?

Answer: *Example*: Self driving cars would need to see the road like your eyes do. (answers will vary)

Q: What actuators do humans have to act on the world around them?

Answer: Arms, legs, fingers, muscles, etc.

Q: How might these actuators and sensors combine (either on a human or robot) to *create a more complex movement or action?*

Answer: *Example*: Our nose might smell something we want to walk towards, but we still have to use our eyes to make sure we aren't bumping into anything on the way. The input from our nose and eyes influence how and where our legs move to walk us towards the smell. (answers will vary)

Q: What are the controllers that humans have to combine input and control action?

Answer: The brain, spinal cord, control loops (handle reflexes, homeostasis, blood flow etc. based on outside information without having to conciosly think about it)

2) Main Lesson

Recommended: 35 minutes/hours

Go through each of the Sensors, Actuators, and Controllers in the drone kit while having students take out each part and lay them out on their desk.

➡ Students can use this graphic organizer to fill in the name, type, and purpose of each part.

Note: The student book has simple definitions of drone parts to get a basic understanding. Everything will be described in greater detail later in the book.

✔ Exercise: While discussing propellers in the actuators portion, ask your students to identify which of their propellers are clockwise and which are counterclockwise using the arrows. Have them think critically about which propellers would have to be attached to which arm and spin when trying to preform different flying tasks (ex. flying up and straight, flying to the right without losing hight, etc.). If helpful, students could use this diagram to draw on and visualize the motion of the motors and propellers.

Note: The way the propellers spin to maneuver the drone the way we want them to is actually pretty complicated. If you want to get into it in full detail with your students, there are four short youtube videos linked down below that explain flight in more detail.

3) Ending The Lesson

Recommended: 10 minutes

- Make sure that your students can identify every part of the drone, so they aren't confused on vocabulary during the build.

✔ Exercise: The teacher can call out the name of a part and see which student can grab theirs first. You could also have them sort all of their parts into groups (actuators, sensors, controllers, and other) on their desk.

✔ Exercise: The teacher could also hold up a part or show a picture of a part on the screen, and the students could raise their hand to say (or write down):

1. The name of the component (no acronyms)
2. Whether it is an actuator, sensor or controller
3. What that device does
4. How it functions (might not be covered indepth)

5. What does it interface with

Note: Teacher should make sure the students understand that all of these components are just building blocks that can be put together to build a drone, but that can also be used in robotics to accomplish a multitude of tasks.

☀ Journal activity: Have your students come up with another use for a part or combination of parts that would allow a robot to do something other than fly a drone. Encourage them to use the vocabulary from this lesson in their responses.

Useful Resources and References

1. Explanation of Drone Flight Dynamics
2. Omnicoptor Flight Dynamic Video
3. Flying Machine TED Talk Video
4. Aggressive Flight Video
5. Quadcopter TED Talk Video

UNIT A.3.2
Safety

Student version (unknown ref duckiesky_high_school_student/introduction-operation-safety)

previous **warning** next (8 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/introduction-operation-safe-
ty'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

KNOWLEDGE AND ACTIVITY GRAPH

Requires:

Hardware - Completed drone build if available, or use video resource to demonstrate the pre-flight safety check.

Previous lesson - N/A

Results:

Knowledge -

- Students will be able to know the laws regulating hobby drone flight.
- Students will be able to identify a safe environment for flight.
- Students can identify potential hazards of drone flight and soldering.
- Students will also learn about proper safety procedures to minimize safety hazards.

Skills - Students will learn about the purpose of and be able to find the pre-flight safety checklist.

2.1. Safety

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

ISTE: 2. b.: Engage in positive, safe, legal and ethical behavior when using technology, including social interactions online or when using networked devices.

NGSS: HS - ETS1 - 3: Evaluate a solution to a complex real-world problem based on prioritized criteria and trade-offs that account for a range of constraints, including cost, safety, reliability, and aesthetics, as well as possible social, cultural, and environmental impacts.

2) Assessments and Evidence of Understanding

By the end of this lesson, students will recognize that there are laws regulating hobby drone flight in their area, and they will be able to identify a safe environment that they can fly their drones in. They will be able to identify potential hazards of drone flight and soldering, and will learn about proper safety procedures to minimize risk of these hazards.

Students will complete an assignment, that they can answer the questions and hand in to their teachers as homework.

3) AGENDA (Brief Summary of Activities)

5 min: Introduction to the lesson.

30 min: Students will analyze a case study, learn about FAA rules, be able to recognize a safe space for flight, learn about safety hazards and the pre flight checklist.

20 min: Ending of lesson by preparing and flying drone demonstration

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Student Materials:

- Drone and soldering kit
- Safety equipment

Teacher Materials:

- Drone and soldering kit
- Safety equipment
- Projector
- Print out the FAA rules handout, home assignment, pre-flight checklist handout

Classroom Set Up

- Have an area of the classroom free for teacher example drone flight and neighborhood modeling activities.

2.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

Hook: The drones that students are building is relatively powerful, and can cause harm to them or others if not used safely.

- Can show a drone fail compilation video and note the danger in those videos
    - Example video:Video involves mostly crashes from bad locations of flying

2) Main Lesson

Case Study: The Midair Collision in 2009

- Do case study about the Midair Collision in 2009 where safety was compromised
  - Collision between a private airplane and a sightseeing helicopter over Manhattan that resulted in 9 deaths.

  ☀ EdX lectures on the Midair Collision

- Video of impact druing incident
- Animation by NTSB explaining the incident
- Explain the causes that contributed to this:
  - The limitations of the see and avoid concept
  - Teterboro Airport local controller's non pertinent telephone conversation
  - Inadequate FAA procedures and regulations
- Explain what the NTSB is, annd what are NTSB reports
  - NTSB: National Transportation Safety Board
  - NTSB reports: provides detailed accounts of accidents.

  ☀ The NTSB report of the collision

  ☀ article 1 and article 2

✔ Have a class discussion on the importance of this event.

- Examples of questions that can be considered during the discussion:
  - How might the see and avoid concept be applied to the smaller drones we are flying?
  - What are some distractions that you might experience when flying your drone?
  - How can we mitigate these distractions?
  - Some of you may drive cars. What safety skills from driving can be translated to flying?

FAA rules

- Explain what FAA stands for and what it is responsible for
  - FAA: Federal Aviation Administration
  - responsible for regulation of civil aviation: including airports, air traffic management, certification of people, certification of aircraft, and protection of US assets.

  ☀ FAA website

- Students must follow FAA rules when operating their drones
- Go over some of the general important safety guidelines by the FAA, the most applicable rules for students are in the handout.

Note: Students are considered to be recreational users.

✔ Exercise: Teachers can set up a diorama of a mini neighborhood in the classroom (with paper or boxes). Teachers can hold a paper airplane over it, and ask students to guide the "drone" through the neighborhood. Teachers can also ask students where and where not they can fly and their reasoning.

Where to fly:

- Explain to students where/how they can choose a space where they can safely operate their drones.

> Note: Encouraged to fly your drone indoors if you have enough space. FAA rules do not apply to operations that take place indoors.

- If flying outdoors, ask students to double check their airspace and restrictions/regulations in their area:
  - FAA's Aeronautical Chart Users' Guide
  - the B4UFLY app

Possible Sources for Danger:

- Explain there are potential sources of danger and students should be constantly aware when operating their drone.

✓ Exercise: While teachers list potential sources of danger, students can pick out drone/soldering kit part that could present that danger. Students should also name the part(s).

- Explain potential sources of danger with soldering:
  - extreme heat and possible burns
  - don't leave materials unattended
- There are several possible sources of danger that can result from the drone:
  - applying force to your body
  - energy discharge from the body
  - parts or propellers dislodging from the drone
  - electric shorts and fires
  - faulty battery

Safe Environment

- Explain "The Bystander Effect" and emphasize that students should always be aware of surroundings and be ready to act
  - the more people that are present, the less likely someone will help a victim during a situation
- Explain that students should always make sure that have a safe environment to fly
- If students can solder or fly their drones in a designated school space, please make sure to address any rules or protocol that must be followed to ensure that students are safe or in case of an accident.
- Explain important safety equipment:
  - Safety glasses, gloves, walls, distance
  - net (not required but point it out if there is one in the class)

✓ Exercise: Get students to gather their required safety equipment while teacher sets up a safe flying space.

3) Ending The Lesson

Pre-flight Checklist

- Introduce the idea of the pre-flight checklist, and that students must go through it whenever they fly

✔ Exercise: Teachers can go through the pre-flight safety checklist as they are gearing up their drone flight for the students to see.

- Go through the pre-flight checklist that is on the handout:

> Note: Teachers should have the pre-flight checklist handout printed for each student, and make sure they always have it with them during flights.

First Flight:

- Emphasize that teachers should supervise students for their first flight.
    - There are many potential problems that could occur with the first flight due to incorrect build or just it being the students' first experience controlling the drone.
- Go through some situations students may experience/should be aware of, students should recognize them and identify potential ways to fix them.
    - Written on the pre-flight checklist handout

Assignment for students:

✔ Take home assignment for students.

The goal of this assignment is to ask students to think critically about how to ensure robots are operated safely, and to devise guidelines for operating their robot safely.

Section B
Electronics

The goals of this section are to cover the basics of electronic circuits, and to teach students how to solder. Depending on the technical background of your students, some, or all of this section may be skipped. For example, if students already know Ohm's Law, they may wish to skip the first two lessons in the circuitry subsection, but may be interested in the last lesson.

SUBSECTION B.1
Circuitry

This subsection includes analogies and hands-on experiments to convey the concepts of circuits. In the first lesson, students create a circuit to light up an LED. The concept of resistance is taught by having students dim the LED by increasing the resistance in their circuits. The second lesson provides the theory behind the experiments in the first lesson. The final lesson advances the ability of circuits from turning on lights and powering up the drone to carrying signals that contain information to help the drone fly.

UNIT B.1.1
Simple Circuits

Student version (unknown ref duckiesky_high_school_student/electronics-circuitry-simple)

previous **warning** next (9 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/electronics-circuitry-sim-
ple'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

KNOWLEDGE AND ACTIVITY GRAPH

Requires:

Hardware -

- 9V Batteries
- 9V Battery clips
- Alligator clip leads (4 per student)
- LEDs
- Various Ohm Resistors

Previous lesson - N/A

Results:

Knowledge -

- Introduction to the concepts of:
- charge
- voltage
- current
- resistance

Skills -

- Ability to construct a simple DC circuit

1.1. Simple Circuits

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

ISTE 4. d.: Exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems.

NGSS HS-PS3-5.: Develop and use a model of two objects interacting through electric or magnetic fields to illustrate the forces between objects and the changes in energy of the objects due to the interaction.

2) Assessments and Evidence of Understanding

By the end of this lesson, students should be able to construct a circuit with lighted LED, and with brightness that can be varied.

3) AGENDA (Brief Summary of Activities)

5 min: Hook with balloon example

15 min: Explaination of static and current electricity

30 min: Hands-on circuitry activity

10 min: Share-out on build process

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

Activity recommended to be done in pairs. Teachers should pair students to support each others needs.

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Teacher materials

Balloons, teaching method for static and current electricity

Classroom Set Up

May be helpful to have materials already on students' desks.

1.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

> **Recommended:** 5 minutes

Hook:

- Teacher demonstrates the balloon static electricity experiment either on themselves or a student volunteer. Explain why the balloon sticks to the wall. (Teacher's discretion to do the experiment in class, or simply review it).
- The main purpose of this is to introduce electricity and how it interacts with objects in a classic experiment. This lesson will go deeper into transfering electricity purposefully through other means than a balloon.

2) Main Lesson

> **Recommended:** 15 minutes

- Use this image, or one of your own to compare electricity to water flowing through pipes using a pump.

- Indroduce the topics of resistance and voltage.
- Discuss how to get a current to move using different energy conversions (battery, generator, etc.).
- If time, discuss how different materials conduct electricity differently (Conductors vs. Insulators).

☀ Provide students with this diagram of the circuit, as well as the actual resources from the student book.

Note: Warn students not to connect LED straight to battery without the resisitor in the circuit.)

✔ Exercise: Let students trial and error their first build attempts. Ask students to try different resisitors, reverse the resistors, and reverse the LED and see what happens. Teacher should also show this video as an example of an LED burning out.

3) Ending The Lesson

Recommended: 10 minutes

✔ Exercise: Group Discussion: Teacher leads discussion based on the following questions:

Q: "What did you learn?"

Q: "What affect did the resistors have? What happened when you reversed their order?" (Follow up with a brief description of polarity in circuits)

Q: "Did anything go wrong?"

Q: "What do you still want to know about circuits?"

Useful Resources and References

1. Water Pipe Analogy
2. Ohms Law Animation
3. Static Electricity Balloon Resource - scienceworld.ca/resource/static-electricity/
4. Circuit Image
5. Glossary

UNIT B.1.2
Voltage, Current, Resistance

Student version (unknown ref duckiesky_high_school_student/electronics-circuitry-voltage)

previous **warning** next (10 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/electronics-circuitry-volt-
age'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

KNOWLEDGE AND ACTIVITY GRAPH

Requires:

Hardware -

- LED
- Various Resistor
- 9V battery (1/student) (not portable battery of drone)
- 9V Battery clips
- Alligator clip leads (4/student)
- Wire
- Multimeter (1/student)

Previous lesson - Simple Circuits

Results:

Knowledge -

Understanding of the relation of:

- charge
- voltage
- current
- resistance defined by Ohm’s law
- causes and effects of short circuit
- difference between AC and DC

Skills - The ability to build and measure simple circuits using a multimeter

2.1. Voltage, Current, and Resistance

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

NGSS HS-PS3-1.: Create a computational model to calculate the change in the energy of one component in a system when the change in energy of the other component(s) and energy flows in and out of the system are known.

This is a helpful video that shows Ohm's Law through a similar experiment done here.

2) Assessments and Evidence of Understanding

By the end of this lesson, students should have built a circuit and records voltage, current, and resistance, and they have shown that Ohm's law holds for multiple circuits.

3) AGENDA (Brief Summary of Activities)

5 min: Intro to Multimeter

40 min: Circuit Activity

5 min: Clean up

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

Activity recommended to be done in pairs. Teachers should pair students to support each other's needs.

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Teacher Materials

Presentation device for explaination of topics

Classroom Set Up

Might be helpful to have supplies already divided up for students prior to the beginning of class.

2.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

> **Recommended:** 5 minutes

Hook:

- Put students into pairs. Give each pair a multimeter and a battery and give them a couple minutes to answer these two questions:

Q: What do you know about this device?

Q: What questions do you have?

Let them play around with it. Then, let them share out answers as a class.

2) Main Lesson

Recommended: 40 minutes

1. **Build and measure first circuit**: Give the students materials to build a circuit similar to last class's except without the LED. Show them how to measure voltage, resitance, and current with the multimeter.
2. **Explaination of Ohm's Law**: Ohm's law states that the current through a conductor between two points is directly proportional to the voltage across the two points. Show how voltage, resistance, and current relate mathmatically using Ohm's law. (A graphic may be helpful, similar to this one). Explain how "short circuit" is a low resistance, unintended, alternate path through the circuit, and how that could damage components in the circuit, power source, start fire, etc...
3. **Build and measure second circuit**: Make the same circuit except for a different resistor, and ask students to prove Ohm's law using the multimeter.
4. **Build and measure circuit with LED**: Make the same circuit except add an LED, and ask students to prove Ohm's law. (It won't.)
5. **Explaination of why LED doesn't observe Ohm's law**
6. **Explaination of AC and DC**: Explain the difference between AC and DC. Measures AC from a wall outlet using a multimeter.

☀ If time: Brief overview (no math) of other passives (e.g. capacitors resist change in voltage, inductors resist changes in current), actives (controlling flow, diodes as one-way valves, transistors as switches, mention that you can do logic/calculations by combining them), and "integrating circuits" into ICs, assembling those into computers, robots, etc.

3) Ending The Lesson

Recommended: 5 minutes

Clean Up!

✔ Here's a vocab worksheet for in class or for homework.

Useful Resources and References

1. Ohm's Law Explanation
2. How to Analyze Resistive Circuits Using Ohm's Law
3. Electical Circuits and Water Pipes Analogy
4. Glossary

UNIT B.1.3

Signals and Connections

Student version (unknown ref duckiesky_high_school_student/electronics-circuitry-signals)

previous **warning** next (11 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/electronics-circuitry-sig-
nals'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

KNOWLEDGE AND ACTIVITY GRAPH

Requires:

Hardware -

- Uncompleted drone kit

Previous Lesson

- Simple Circuits
- Voltage, Current, and Resistance

Results:

Knowledge -

- Ability to identify different electrical connections used on the drone and the advantages of each
- Know the definition of signals and where they are used on the drone
- Understand the differences between and varying advantages of analog and digital signals
- Understand that values, properties, and concepts can be encoded into numbers and the purpose thereof, and know advantages of binary as a number system.

Skills - N/A

3.1. Signals and Connections

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

ISTE: 1. d.: Understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

2) Assessments and Evidence of Understanding

By the end of this lesson, student should be able to understand analog and digital signals and examples of each, as well as be abe to translate numbers into binary.

3) AGENDA (Brief Summary of Activities)

- 5 min: Introduction
- 20 min: Electrical Signals
- 10 min: Analog signals
- 20 min: Digital signals
- 15 min: Short online game

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Teacher Materials

Optional presentaion medium for lesson (i.e. whiteboard, powerpoint slide).

3.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing the Lesson

Recommended: 5 minutes

Hook:

- The teacher should open this lesson with an explanation that there are two types of signals on the drone (Analog and Digital).

✔ Exercise: Have the students guess where these signals could be located on the drone. If no one is able to give an answer, lead into the lesson saying that these signals are not only intigral to the drone's flight, but also integral to know in the field of robotics.

2) Main Lesson

Recommended: 50 minutes

Electrical connections (20min)

Briefly explain: crimping, connectors, splicing, and soldering. Introduce PCBs as *'home bases'* for connections in the drone.

✔ Exercise: Point out soldered connections and pull out different connectors used in the drone (e.g. from battery sense to Pi GPIO to XT-60), compare the number of pins, current capacity, etc.

☀ Use this section as an opportunity to motivate why students need to learn to solder. This will transition the course to the next module, Soldering!

Analog signals (10min)

Besides power, electricity can also be used to convey information...Information can be transmitted through electricity:

1. Encoding in *voltage*: use an IR sensor as an example.
2. Encoding in *frequency*: use WiFi as an example.

> **Note:** *Noise*, which can come from various sources, can distort or block the information transmittion done by the former.

✔ Exercise: After describing frequency and voltage, have a **group brainstorm** on what other electronics and sensors could have a voltage or frequency signal. (Use examples from above to lead the discussion).

Digital signals (20 min)

Introduce how numbers can represent anything, e.g. letters (a=1, b=2, etc...)

✔ Exercise: Devise a way to represent things as one or more numbers! For example, physical properties (distance, colors, volume, etc...), other symbols, music (notes, chords, progressions, pieces, scales/modes, intervals, etc.), sports plays, games (cards, stats, etc.).

Our number system uses base-10 (10 symbols) to represent numbers, but there are many other ways to convey information!

There's no reason we have to use 10 symbols, you can get away with just 2 – binary! (or other numbers, eight: octal, sixteen: hexadecimal) Show ascii table with binary and decimal.

✔ Exercise: To understand binary, here is a short activity to do with this video explaination. Assign the students to convert a few numbers of the teacher's choosing on their flippydo.

Using the resources below and in the student book, teachers should discuss:

- How digits can be represented as ranges of voltages with an undefined zone in between (used as a break). Also, mention having only two symbols (binary) simplifies the circuitry and reduces effects of noise.
- Binary can be transmitted in parallel and over time, which are two ways to represent more than one symbol, sacrificing either complexity or time.
- Mention electrical vs timing vs content specifications of signals (discussing this will set up for future networking lessons with higher levels of abstraction).

3) Ending The Lesson

> **Recommended:** 15 minutes

✔ Give students this binary game link, and challenge them to hit the highest level!

Useful Resources and References

1. Flippy Do Activity
2. Binary Bonanza Game
3. ASCII Table
4. Glossary

Subsection B.2 Soldering

This subsection contains an introduction, practice, and troubleshooting techniques to build strong soldering skills for the students. Students' abilities to to solder well, identify soldering errors, and fix mistakes is essential for a successful drone build. The more adept students are, the less fixing they will need to do later on. The time required for students to really get the skill will vary; feel free to add or remove soldering exercises as needed for your particular group of students.

Unit B.2.1
Intro to Soldering

Student version (unknown ref duckiesky_high_school_student/electronics-soldering-introduction)

previous **warning** next (12 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/electronics-soldering-intro-
duction'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

Knowledge and activity graph

Requires:

Hardware -

- Drone kit
- Soldering iron
- Soldering iron cleaning supplies (golden curl soldering cleaners and sponges)
- Solder
- Wire strippers
- Wire cutters
- Working space
- Helping hands
- Long-nose pliers
- Eye protection
- Soldering fan
- Other school mandated safety equipment

Previous lesson - N/A

Results:

Knowledge -

- Identifying wire gauges

Skills -

- Stripping wires
- Soldering iron to tin the wires

1.1. Intro to Soldering

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

NGSS: HS-PS3-3: Students who demonstrate understanding can design, build and refine a device that works within given constraints to convert one form of energy into another form of energy.

ISTE: 1d: Understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

2) Assessments and Evidence of Understanding

By the end of this lesson, students will be able to use the correct wire stripper gauge for each wire to successfully strip and tin the ESC, Motor, BEC, Red and Brown Wire.

3) AGENDA (Brief Summary of Activities)

1 min: Hook

54 min: Intro to Soldering Lesson

5 min: Clean Up and Optional Exit Ticket

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

Students with dexterity problems. Inclusion might be teacher/student assistance, helping other students with gathering material, cleaning, taking care of equipment etc.

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Materials needed

Teacher Materials:

Screen to play video (if desired), examples of correctly cut and tinned wires

Classroom Set Up:

Teacher can write a DO NOW on the board for students to grab their drone kits and sit at their maker's space.

1.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

> **Recommended:** 1 minute

Hook:

- This will be students' first step to building their own drone. It is important to learn and practice basic soldering skills here so that they can feel confident when doing more advanced soldering work later on on the drone.

☀ It is important to prepare the grunt work so that their drone building is easier and faster.

☀ Working here and knowing how to use maker's tools will give students the knowledge and tools to embark on other electrical/maker projects.

2) Main Lesson

Recommended: 54 minutes

Wire Gauges and Stripping:

- Reintroduce the 4 ESC, 4 Motor, 1 BEC and 1 Red and Brown wire pair found in the flight control box and where to find in students' drone kit. Students should gather these materials and set them on the side.
- Take one of them (that is pre tinned) and point out where to find the wire gauge.

Note: The gauge is written on the wire coating. It will be a number followed by AWG (American Wire Gauge). A pre tinned wire is a wire tinned at the factory. It looks shiny on the tip and the end cannot be frayed.

- Take a wire stripper and show where the wire stripper displays its AWG number. Then take the model wire and strip it.

☀ You may also recommend that students use one level lower strip number if it does not work so that the wire strips easily and does not rip the internal wires.

- Take the wire cutter and show students a pre tinned wire. Now use the wire cutters to cut the pre tinned part off. Emphasize that all wires that are pre tinned must be cut off and re tinned.

✔ Exercise: Teachers now have students strip, cut, and tin the same wire as they did and answer questions.

- Explain what soldering and tinning is and the safety components of soldering.

✔ Exercise: Lead students through or demonstrate the process of setting up their soldering station (preheading and cleaning the iron before beginning), turning on the soldering iron, and cleaning the iron with gold curl soldering cleaners and a sponge.

Note: Each soldering iron model is different and you should know an appropriate heat to tin with.

✔ Exercise: Demonstrate how to tin using the example wire, and then have students try to tin their own wires using the instructions found in the student book or the teacher's own instruction.

Note: The teacher should emphasize safety during this tutorial. You may use this video (1:30-3:00) before or after your tutorial to emphasize how to tin.

✔ Exercise: Circulate assisting students in tinning all of the required wires.

All wires for the following components must be self tinned:

1. 4 ESCs

2. 4 Motors
3. 1 BEC on the IN side
4. Red and Brown Wire Pair in the Flight Control Box

3) Ending The Lesson

> **Recommended:** 5 minutes

- Have students clean up their work space, using the appropriate methods and safety precautions.

✔ Optional Exit Ticket: Students will reflect and write down what they have completed and what they potentially still need to do for the next class.

Useful Resources and References

1. Detailed instructions of each component and tinning
2. Soldering Tutorial for Beginners in 5 easy steps
3. How to Tin a Wire (in lesson)
4. Glossary

Unit B.2.2
Building Skill

Student version (unknown ref duckiesky_high_school_student/electronics-soldering-skill)

previous **warning** next (13 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/electronics-soldering-skill'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

Knowledge and activity graph

Requires:

Hardware -

- Soldering irons
- Soldering mats
- Solder fume extractors
- Solder rolls
- Solder fluxp
- Spare wire

Previous lesson - [Intro to Soldering][#electronics-soldering-introduction]

Results:

Knowledge -

- Understanding the principles of effective soldering

Skills -

- Being able to through-hole solder succesfully

2.1. Building Skill with Soldering

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

NGSS: HS-PS3-3: Students who demonstrate understanding can design, build and refine a device that works within given constraints to convert one form of energy into another form of energy.

ISTE: 1d: Understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

2) Assessments and Evidence of Understanding

By the end of this lesson, students should be able to solder with flux and through-hole solder.

3) AGENDA (Brief Summary of Activities)

5 min: Intro

50 min: Tips on how to best use soldering iron + Experiment

5 min: Regroup and Discussion

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

Work in pairs, but if possible have one soldering unit for each student. Teacher should create these pairs to cater to each student's needs.

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Teacher Materials

Should have a set of student materials for themselves to demonstrate to students.

2.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

> **Recommended:** 5 minutes

Teacher should demonstrate a successfully executed soldering of wires using flux to create a circuit.

2) Main Lesson

> **Recommended:** 50 minutes

Teacher should discuss and lead discussion on:

- Oxidation and how it affects the longevity of solder connections (What are other examples of oxidation? Why then should we try to prevent oxidation?)
- Flux, and how applying it to connection points before soldering can make for easier application of solder ("What do you think the soldering experience with flux versus without will be?")
- How different temperatures of the soldering iron can affect how well the solder applies

These topics are very relevant to the construction of the drone, since it is impertive that the soldering is well done and strong, with no oxidation that could mess up a circuit.

PCBs (Printed Circuit Boards) and PCB delamination is also imporant to discuss. Delamination occurs when too much heat is applied to the board, and bubbles and cracks form. That is why you shoudl never keep the soldering iron on a PCB for and extended period of time, as it can ruin the component. Here's an in-depth video about PCB delamination, but the explaination here is suitable for most classroom needs.

If there's time through-hole soldering is another form a soldering used on the Pi on the drone. This is simply when a wire is stuck through a previously made hole on a PCB. To complete through hole soldering, tin the wire, stick it through the pre-made hole, and solder the tip of the wire to the metal surrounding the hole.

Student Experimentation

Give the students the tools in the materials section (spare wire, soldering tools, and flux), and ask them to solder wires together with and without flux, while also showing them how to get acclimated to using the various materials. This half hour should be dedicated to students becoming familiar and comfortable with using the soldering iron combined with flux.

To solder with flux, simply follow trhe steps of soldering from the previous lesson, except before tinning the wire, stick the soldering iron into the flux and apply it on the wires. This will make the solder flow easier, but is not always necessary.

Teacher should check student's soldering joints as the students are working.

Heads up for next lesson: Keep all soldering joints that are weak or problematic, they will be useful in the next lesson.

3) Ending The Lesson

> **Recommended:** 5 minutes

Teacher should ask the students, on a scale of 1 to 10, how comforable they feel with the tools. If the majority of the class still feels uncomfortable, it might be wise to take half of another class to continue to work on their skills.

Useful Resources and References

Soldering with Flux - https://www.youtube.com/watch?v=3Z8CzB4BYJA

Glossary

Unit B.2.3
Troubleshooting

Student version (unknown ref duckiesky_high_school_student/electronics-soldering-troubleshooting)

previous **warning** next (14 of 36) index
warning
```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duckiesky_high_school_student/electronics-soldering-troubleshooting'.
```
Location not known more precisely.

Created by function `n/a` in module `n/a`.

Knowledge and activity graph

Requires:

Hardware -

- Soldering iron
- Soldering mat
- Solder fume extractor
- Solder roll
- Solder flux
- Spare wire
- Proto board

Previous lesson

- [Intro to Soldering][#electronics-soldering-introduction]
- Building Skill

Result:

Knowledge -

- Understanding of the causes of soldering failures
- Knowing techniques for avoiding soldering mistakes

Skills -

- Identifing and fixing soldering mistakes
- Being able to do wire-to-pad soldering

3.1. Troubleshooting Soldering

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

NGSS: HS-PS3-3: Students who demonstrate understanding can design, build and refine a device that works within given constraints to convert one form of energy into another form of energy.

ISTE: 1d: Students: understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

2) Assessments and Evidence of Understanding

By the end of this lesson, students will be able to solve common soldering errors.

3) AGENDA (Brief Summary of Activities)

5 min: Intro

40 min: Problems and Solutions for Soldering

5 min: Ending the lesson

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

Activity recommended to be done in pairs. Teachers should pair students to support each other's needs.

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Teacher Materials

Should have a set of their own to demonstrate to students (or simply demonstrate on a student's kit).

3.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

> **Recommended:** 5 minutes

Hook:

- This lesson is necessary for students to have a grasp on how to troubleshoot problems with their soldering. Teachers should display mitsakes they've made in soldering the wires on their drone, and demonstate/explain how they fixed the problem.

2) Main Lesson

> **Recommended:** 40 minutes

The teacher should present an array of problems such as (but not limited to):

- Cold joints
- Joints with too much solder

- Joints that don't flow properly

Some troubleshooting solutions a teacher may present are:

- Inspecting all joints carefully
- Cleaning
- Using the multimeter
- Testing strength of joints with fingers

✔ Exercise: Students should use materials listed above and have partners create bad soldering jobs (too much solder, bad connection), and have their partner diagnose and troubleshoot. More importantly, the students should focus on how these soldering mistakes. The teacher should walk around the class, making sure the student's aren't causing any harm to the materials unnintentionally. The teacher could also give out some of the student's past soldering jobs that failed, and possibly ask the whole class to diagnose it together.

3) Ending The Lesson

Recommended: 5 minutes

Clean up! This practice will use a lot of materials, so be sure to have students clean up their workspaces.

Useful Resources and References

1. Glossary

Section C
Build: 1

Student version (unknown ref duckiesky_high_school_student/build1)

previous **warning** next (15 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/build1'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

The goal of this build part is to get power to the Raspberry Pi so that the students can use it to learn to program. Students will make the circuit connections using the soldering skills that they gained from the Soldering subsection. By the end of this build part, students should be able to power up their Pis and connect to them on their base stations.

SECTION D
Computing and Networking

The goal of this section is to introduce students to essential technical skills for robotics. Networking allows student's base stations' to talk to their drones. Bash allows students to run programs using just a text interface. Combining the knowledge of networking and bash, students can do cool things on the drone, such as make an LED blink, and eventually make their drones fly.

SUBSECTION D.1
Using the Pi

This subsection includes an overview of Networking, SSH, Bash. After completing this subsection, your students will be able to connect to the Pis, navigate in Bash, and blink its LED using a command line interface.

UNIT D.1.1
Networking

Student version (unknown ref duckiesky_high_school_student/computing-pi-networking)

previous **warning** next (16 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/computing-pi-networking'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

KNOWLEDGE AND ACTIVITY GRAPH

Requires:

Hardware - Basestation
Previous lesson - Build Part 1

Results:

Knowledge -

- 7 layers of network abstraction
- IP and MAC addresses
- Definition of the computer terminal/shell
- Definition and use of SSH

Skills - Basic Text Editor and VSCode skills

1.1. Networking

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

ISTE: 1. d.: Understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

2) Assessments and Evidence of Understanding

By the end of this lesson, students should be able to connect to the Pi and verify its the right one by running an existing bash script to blink the built-in LED.

3) AGENDA (Brief Summary of Activities)

10 min: Basestation Setup

30 min: Networking Lesson

25 min: Pi Connection and Led Blinking Script

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Teacher Materials:

Basestation, a projector (optional)

Classroom Setup:

Teachers can write a DO NOW on the board for students to set up their basestations.

1.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

> **Recommended:** 10 minutes

Hook:

- This will be a lesson for the student on the basics of Networking and Computer Terminals. It is important to comprehend networking to understand the basis of how computers connect with each other and to know computer terminal details to understand how computers can be controlled.
    - One robot can be made up of one or more computers that need to coordinate with each other, and robots may need to communicate with external systems; for these purposes, networking is essential.
- (Optional) Networking should be applied to how the drone communicates with the basestation. To do this, teachers can start class with a small discussion instead of or in addition to the above hook. The following questions are potential discussion points:

Q: What are some other robots or devices in your daily life that have to communicate with each other? How do you think they do this?

Q: How will we control our drone? From what?

Q: What features of the communication channel between our basestation and the drone could be important? Wireless connectivity, latency (delay between communications), throughput (maximum volume of communications)?

2) Main Lesson

> **Recommended:** 40 minutes

- Teachers should cover the basics of the 7 layers of network abstraction and IP/MAC addresses as outlined in the student book.
- Teachers define (and potentially demonstrate) what a computer terminal and shell is as outlined in the student book.
- Teachers define SSH and what it is used for as outlined in the student book.

3) Ending The lesson

Recommended: 10 minutes

✔ Exercise: Teachers should have students connect to the Pi via web browser and explain each of the components of the text editor.

✔ Optional Exercise: Teachers can attempt to follow our old build instructions to connect to the drone over SSH; if you are able to, you can try to have students follow the similar process. These instructions are not too detailed and are outdated, so it may take additional troubleshooting on your end to figure this out (additionally this may not be compatible on chromebook).

Useful Resources and References

1. Networking edX Lecture
2. Shell and Terminal Details
3. SSH Details

UNIT D.1.2
Bash

Student version (unknown ref duckiesky_high_school_student/computing-pi-bash)

previous **warning** next (17 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/computing-pi-bash'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

KNOWLEDGE AND ACTIVITY GRAPH

Requires:

Hardware -

- Basestation
- Drone (Build Part 1 completed)

Previous lesson - Networking

Results:

Knowledge -

- Computer file structure
- Essential bash commands

Skills -

- Navigating computer directories in the terminal
- Editing files through the command line

2.1. Bash

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

NGSS: HS - ETS1 - 2: Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.

ISTE: 1. d.: Understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

2) Assessments and Evidence of Understanding

By the end of this lesson, students should be able to complete the below Bash terminal task.

3) AGENDA (Brief Summary of Activities)

5 min: Basestation Setup

35 min: Bash Lesson

20 min: Bash Terminal Task

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Teacher Materials:

Basestation, a projector (optional)

Classroom Set Up:

Teachers can write a DO NOW on the board for students to set up their basestations.

2.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

> **Recommended:** 5 minutes

Hook:

- This will be a lesson for the student on Bash and the Shell. It is important to learn how to utilize a shell as it is the introduction to the inner processes of the operating system. Additionally, the shell will be our gateway to interacting with the Pi on our drone.
- Conceptually connect Bash to Markdown in the sense that the computer is being affected via text: the purpose of Bash is to "control" the computer vs whereas Markdown is utilized for the purpose of "visual layout."

2) Main Lesson

> **Recommended:** 40 minutes

Teachers can either use a live demonstration of the exercise via a projector using the webpage for the Pi's text editor where students follow along, let students follow along with the student book instructions, or utilize an online Bash tutorial (see *Useful Resources and References* below) that the students can work through as an alternative.

- (Optional) Have the teacher check off the outputs in the terminals for the exercise

By the end of the Main Lesson, students should be able to:

- See basic information in the terminal and navigate the file system (pwd, ls, cd).
- Creating and removing files and directories (touch, rm, mkdir, rmdir).
- Using a carat to output to a file (e.g. "echo 'foo' >bar.txt").
- Printing output in the terminal (echo, cat).
- Clearing terminal output and exiting (clear, exit).

3) Ending The Lesson

Recommended: 15 minutes

✔ Exercise: Students should do the exercise in the textbook with Bash commands to test out their knowledge of the terminal. They can do this through the online web editor when connected to the Pi.

Useful Resources and References

1. Shell Tutorial 1
2. Shell Tutorial 2
3. Bash Command Overview
4. Bash Shell Script Tutorial

UNIT D.1.3

Blinking an LED

Student version (unknown ref duckiesky_high_school_student/computing-pi-led)

previous **warning** next (18 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/computing-pi-led'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

KNOWLEDGE AND ACTIVITY GRAPH

Requires:

Hardware -

- Basestation
- Build Part 1 completed

Previous lesson -

- Networking
- Bash

Results:

Knowledge - Definition and purpose of Raspberry Pi GPIO

Skills -

- Read the GPIO pin chart for the Raspberry Pi
- Write a bash script to blink their LED

3.1. Blinking an LED

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

NGSS: HS - ETS1 - 2: Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.

ISTE: 1. d.: Understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

2) Assessments and Evidence of Understanding

Students write a bash script to make the led that they've soldered blink.

3) AGENDA (Brief Summary of Activities)

5 min: Basestation Setup

20 min: Scripts and REPLS Lesson

35 min: Writing a Bash Script

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Classroom Set Up

Teacher can write a DO NOW on the board for students to set up their basestations.

3.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

> **Recommended:** 5 minutes

Hook

Teachers can talk about the following or their own.

- This will be a lesson for the student on creating a bash script that will blink the LED on their drone. It is important to understand scripts as they will come up when continuing with programming.

2) Main Lesson

> **Recommended:** 20 minutes

1 - Teachers cover what a REPL (Read–eval–print loop) is and what a script is

In the bash lesson, we have already learned how to control our computer in the terminal. However, we seem to only able to type one line/command at a time. Can we do better than that? Before we discuss the possibilities, we introduce two related terminologies.

- Read–eval–print loop: a simple interactive computer programming environment that takes **single** user inputs, executes them, and returns the result to the user.
- Script: a programming language for a special run-time environment that automates the execution of tasks; the tasks could alternatively be executed **one-by-one** by a human operator.

    ➡ Difference: script saves the commands to be run sequentially in the future, while the REPL runs only one command at a time. Previously in the bash lesson, all the commands you ran were executed by the REPL. In this lesson, you will learn how to 'save' multiple commands and run them all at once in bash!

2 - Teachers speak on Pi GPIO numbering

Motivation: The goal of this lesson is to our terminal on the Pi to blink the LED soldered on your drone. LED lights up when their is electric current flows throught it. Therefore, we want to write commands that activate the pin on the Pi to provide the current. However, we need to know first how to refer to a specific pin in the terminal.

- On the Raspberry Pi, Pin can be defined in two ways, using the **GPIO Board** and **GPIO BCM** methods.
- **GPIO Board** refers to the naming on the Raspberry Pi board. This is the numbering on the Pi Hat.
- **GPIO BCM** refers to the default pin of the Broadcom SOC channel chip. This is the numbering we are going to use in this lesson. We need to know how the BCM pin numbering corresponds to the numberings on the Pi Hat. Here it is.

3) Writing a Bash Script

Recommended: 35 minutes

✔ Exercise: Instructors show how to make and run bash scripts, delays, printing, pin setting by leading students write a bash script to make the led that they've soldered blink.

See the student book for details

Useful Resources and References

1. More information on REPLs
2. More information on scripts
3. More information on blinking an LED

Section E
Sensors, Actuators, and Control: 1

This section contains the first introduction to sensors to prepare students to add the first sensor to their drone. The goal is to motivate the need for sensors, and to explain their theory so that students understand the reason for adding each component to the drone.

Subsection E.1 Overview

This sections outlines the basic purpose of sensors on robots.

Unit E.1.1
Intro to Sensors

Student version (unknown ref duckiesky_high_school_student/sac1-basics-sensors)

previous **warning** next (19 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/sac1-basics-sensors'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

KNOWLEDGE AND ACTIVITY GRAPH

Requires:

Previous lesson - Signals and Connections

Results:

Knowledge -

- The ability to differentiate between examples of continuous signals and those discretized in time and/or value
- Know the purpose of discretizing continuous signals
- Know the general pathway for measuring a physical system potential
- Get a sense of the limitations of electrical sensors

Skills - N/A

1.1. Introduction to Sensors

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

NGSS: HS - PS4 - 5: Communicate technical information about how some technological devices use the principles of wave behavior and wave interactions with matter to transmit and capture information and energy.

2) Assessments and Evidence of Understanding

By the end of the lesson, students are able to understand how sensors process signals, as well as how they are subject to errors.

3) AGENDA (Brief Summary of Activities)

10 min: Brainstorming the senses of robots based on human experience

25 min: Introduce conecepts about sensors

15 min: Fun exercise to help students experience the concepts introduced

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Classroom Set Up

Teacher can write a DO NOW on the board for students to think about what can be measured by a robot.

1.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

Recommended: 10 mintues

Hook:

✔ Exercise: Students brainstorm and discuss in groups the things that can be measured by a robot, and possibly write their ideas on board.

After this is done, teachers do a review what our drone's three sensors measure:

- IR sensor - measures distance
- IMU sensor - measures acceleration
- The camera - measures planar motion

Explain the definition of roll, pitch, and yaw.

✔ Exercise: Have your students make paper airplanes to physically demonstrate roll, pitch, and yaw.

2) Main Lesson

Recommended: 25 minutes

The main lesson is evolved around introducing 3 questions, possibly let students discuss their solutions, and then provide the linked student book answers.

Question 1. **How computers interpret sensors' output.**

Q: Refer to the signals lesson that computers can only understand a finite set of numbers, so we need to convert sensors' output to a finite set of numbers.

Pause for discussion if necessary

Answer

Question 2. **How we acquire data that are not directly measured.**

Q: What if we want the data in between those signals? Or if we want to predict the future values?

Pause for discussion if necessary

Answer

Question 3. **How good are our sensors?**

There are two measures of the “goodness” of a sensor:

```
1. Accuracy: How close are the measurements to the truth

2. Precision: How close are the measurements of the same item to each
other
```

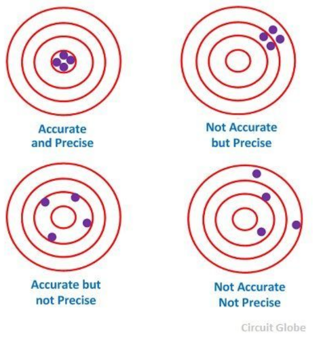

Figure 1.1. Accuracy and Precision Graph

☀ We need to pay extra attention to the units used. Here are the industry standards.

```
- Meters, NOT inches or feet

- [Radians](https://en.wikipedia.org/wiki/Radian), NOT degrees
```

Q: Sensors have errors, too! Just like the signal noise from the electronics unit. How might we mitigate them?

Pause for discussion if necessary

Answer

3) Ending The Lesson

Recommended: 15 minutes

A quick review of the lesson, sample questions include:

Q: What does ADC stand for?

Q: Why do we need an ADC?

Q: Can you demonstrate row, pitch, and yaw using your body?

✔ Exercise: Mimicking the second way (combining data from multiple sensors) that make sensors more accurate – students individually measure something (a distance of a paper airplane flight, number of jelly beans in a jar, or any other creative activities) and then averaging the individual guesses. Then, ask students how accurate their measusures are. What about precision?

Useful Resources and References

SECTION F
Build: 2

Student version (unknown ref duckiesky_high_school_student/build2)

previous **warning** next (20 of 36) index
warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/build2'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

In this part of the build, students will be adding thir first sensor to the drone – the infrared (IR) sensor. The motivation for adding the IR sensor is to measure how high the drone is while flying. By the end of this build part, student's should be able to see the output of their IR sensors in a web browser using the drone's web interface.

SECTION G
Sensors, Actuators, and Control: 2

The goal of this section is to leverage the Infrared Sensor and LED as opportunity to explain the challenges and solutions of using a sensor and controlling an actuator. In particular, in this section you will be guiding students through the processes of reading the *raw* output from a sensor, converting the raw output into a useful measurement, and then using a framework called ROS to share the measurement amongst programs running on the Raspberry Pi. Raw sensor output is some signal that is either a varying voltage or current, and the process of calibration is used to convert the signal into a measurement, such as distance. The steps in this section are applied to all of the sensors on the drone; however, students only look at the process for the IR sensor at first because it is the most straightforward example.

Once the sensor data is calibrated and shared using ROS, the students will create an *open loop controller* to vary the brightness of the LED based on the measurement from the IR sensor. The open loop controller is the simplest form of a control system, and it motivates the need for ROS since there is one program sharing the data (the program that reads the IR sensor) and there is another program using the data (the open loop controller).

SUBSECTION G.1
Sensing

The goal of this subsection is to teach students how to read and interpret the IR sensor output. Although the steps in this subsection are implemented for the IR sensor only, it should be emphasized that the steps can be adapted for any sensor. It is important to show students how to learn to use a sensor on their own so that they can reapply the steps for different use cases.

UNIT G.1.1
Calibration

Student version (unknown ref duckiesky_high_school_student/sac2-sensing-calibration)

previous **warning** next (21 of 36) index
warning
```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/sac2-sensing-calibration'.
```
Location not known more precisely.

Created by function `n/a` in module `n/a`.

KNOWLEDGE AND ACTIVITY GRAPH

Requires:

Hardware -

- Basestation
- IR sensors
- Multimeter
- Graph paper or a computer with Excel

Previous Lesson - Introduction to Sensors

Result:

Knowledge -

- Understanding that sensors don't provide readings in standardized units or scales
- The the specific nonlinearities of the IR sensor
- Equation of a line and its use for linear approximation
- How to interpolate

Skills -

- Calibrating measurements by linearizing the readings and then interpolating known readings

1.1. Calibration

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

ISTE: 5.c.: Collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making.

2) Assessments and Evidence of Understanding

By the end of this lesson, students should have a completed *Voltage vs. 1/distance* graph.

3) AGENDA (Brief Summary of Activities)

25 min: In depth explanation of the IR sensor

25 min: Activity graphing the inputs and outputs of the IR sensor to linearize it and interpolate values using the equation of the line

5 min: Wrap up and explanation of the bigger purpose of this exercise

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Teacher Materials

- Already completed *Voltage vs. Distance* and *Voltage vs. 1/distance* graph to compare with students' graphs. Feel free to make your own, or there are links to example graphs below.
- Projector/ slides to show graphs if needed.

Classroom Set Up:

- Teachers can write a DO NOW on the board that says "Start thinking about what you already know about Infared Sensors. What does that name make you think of?"
- Each student should have a multimeter either on their desk or ready to be passed out, along with their IR Sensor.

1.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

> **Recommended:** 25 minutes

Hook:

- Discuss what students already believe their IR Sensor is (as stated in the DO NOW) and fascilitate a discussion of the importance of monitoring drone height in the right units and communicating that with different parts of the drone. This lesson is important because not only does it help students understand how their own drones' sensors recieve and send vital information, it also reinforces basic graphing and conversion skills that have many applications in the world of STEM and beyond.

Using the student book material or the links listed below, explain what Infared (IR) Sensors are and how they work.

☀ It might be useful to review what infared light is and where it can be found. What environmental factors could potentially alter our drone's IR readings?

Possible discussion question:

Q: Will there be a higher voltage emitted when the sensor is closer to the surface or further away?

Answer: There will be a higher voltage reading when the sensor is closer to the surface, because the infared light bounding back will be stronger the less distance it has to travel.

✓ Exercise: Have your students use the multimeter to measure the voltage from their IR sensors at different distances away from the floor or their desks. This video an example of students doing this. Either on a piece of graph paper or in an Excel spreadsheet, have them record and plot this data with "height above ground (cm)" as the x-axis and "IR sensor voltage (V)" as the y-axis to identify a relationship between the two variables (they will come out to be the inverse transform). Then, they can calculate the distance between the surface and the sensor by using only the voltage.

2) Main Lesson

Recommended: 25 minutes

Analyze the Data

1. Have students analyze their own graphs to try and find a trend in the data. Encourage note taking and group collaboration on this, since they should all have similar data points.

Q: What does the graph show about the uses and limitations of the IR sensor?

1. Pull up your own graph of IR values to show students. Ask your students about various points and see if they compare to yours to show that students' values may be different.

Note: A particular voltage has *at least* two potential distances associated with it. Point out that the sensor stops working past a certain distance.

1. Emphasize that we have to make the assumption that the sensor is within the useful range, and that we have to enforce that it stays within that range when flying, so there will be a height limit when flying the actual drone.
2. Present the _voltage vs distance) graphs in the IR sensor data sheet. The data sheet is like an instruction book for a sensor: it contains information about the sensor written by the designer. By familiarizing students with the data sheet for the IR sensor, the goal is to promote students' self-efficacy to learn about other sensors on the drone and on other robots.
3. Finally, use the linearization shown in the data sheet (voltage vs. inverse distance) to find the actual mathematical relationship between voltage and distance. Highlight that the data takes on the shape of 1/x, and then show a graph of voltage vs 1/distance.

```
Voltage = k/distance + p
```

Figure 1.1. Voltage vs. Distance and Inverse Distance Graphs (Ignore the key)

✓ Exercise: Have your students convert their own plot to be voltage vs 1/distance to see how linear it is. Notice that there's an offset towards the right side of the graph where

1/x is larger (distance to the sensor is smaller).

3) Ending The Lesson

> **Recommended:** 10 minutes

✔ Exercise: Using their linear equations from the above activity, students can now predict Voltage based on a specific Height and vice versa by interpolating within the graph. Put different heights on the board and see who can calculate the Voltage first with the equation, and then have students test it on the actual IR sensor.

- Remind students that even though they used the equation of a line to interpolate and estimate certain voltages and distances, that there are specific nonlinearities of the IR sensor.
- Explain how they calibrated the readings by putting them into a graph so they can estimate data in the middle since sensors don't provide readings in standardized units or scales. We will add an ADC (Analog-to-Digital Converter) to allow our Rasberry Pi to read distance from our IR sensor.

Useful Resources and References

1. EdX lecture
2. IR sensor tutorial

Unit G.1.2
Reading Sensors

Student version (unknown ref duckiesky_high_school_student/sac2-sensing-reading)

previous **warning** next (22 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/sac2-sensing-reading'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

Knowledge and activity graph

Requires:

Hardware - basestation - Build Part 1 and 2 completed - battery

Previous lessons

- Computing and Networking
- Calibration

Result:

Knowledge - Introduction to Python syntax and concepts including global variables, libraries, classes, methods, functions, and while loops

Skills - Write a Python script to read and calibrate measurements from their IR sensor

- Function composition to reuse their calibration function from the previous lesson
- Dimensional analysis to verify that a function produce the desired units

2.1. Reading Sensors with Software

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

2) Assessments and Evidence of Understanding

By the end of this lesson, students will be able to measure distances on their Pi using the IR sensor.

3) AGENDA (Brief Summary of Activities)

5 min: classroom setup 10 min: introduction to the lesson

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Materials needed

Teacher Materials: base station, drone build completed to build part 2, meter stick

Classroom Setup: Teachers can write a DO NOW on the board for students to set up their basestations and open a web browser. Have them complete the creation of a GitHub account process (step 1 in the student book) while you introduce the lesson.

2.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

> **Recommended:** 25 minutes

Activity 1: Reading and Calibrating the Sensor without Programming

This is an interactive activity used to motivate the need for programming.

Divide the class into groups of three. In each group, designate one student to read the voltage value from the multimeter. This student will need to setup their multimeter to read up to 20V. Designate another student to convert the voltage value to a distance in meters. This student will need to a calculator and the calibration formula from the previous lesson. Designate a third student to hold the IR sensor facing down at a surface. This student will need to plug the battery into their drone so the IR sensor has power.

The goal of the activity is to get the IR sensor as close to 0.3 meters as possible. It will require the group to work together to read the IR sensor, convert the reading, and then move the sensor up or down to get the 0.3 meters. While the students are working, write the debrief questions below on the board. After 5 minutes (or more/less depending on the class), the teacher will walk around and check the height of the IR sensor using a meter stick to see which group was closest. If students finish early, they can go on to the debrief questions below:

Have the students debrief by answering the following questions:

Q: What step took the longest? (reading, converting, moving the sensor)

Q: How could you speed up this process?

Q: How did it feel to do the same thing over and over again?

Background:

Explain how sensors need to be read and interpreted quickly in robotics, especially on a drone. The slower we process the sensors' information on the drone, the more time it takes for us (or our computer) to react and control the flight.

✖ Problem: We need an **automated** way of doing the conversion to units. We need measurements many times each second. Automated means that the measurements and conversions happen without the user needing to do anything.

✔ Solution: Write a computer program to read and convert the measurements!

So far, we've seen two coding languages: text editors which format text, and bash which is used for interacting with the computer. We need a coding language whose purpose is "doing computing" (making mathematical computations or calculations quickly). We need Python!

> Note: This is explained clearly in !

2) Main Lesson

> **Recommended:** 60 minutes

Explain that the first challenge with automating the sensor reading is to get the hardware devices to "talk" to eachother. Recall that the IR sensor produces a voltage, which is an *analog* signal. However, many computers, including the Pi, can only read *digital* signals (1's and 0's only). Explain that the ADC takes in the analog signal and converts it to a digital signal that the Pi can read.

Lead the students through Activity 1 in the Student Book. This can be done either by having the students read through the activity, or by following the teacher through the activity on a projector.

Explain that it is inconvenient to keep running the script every time we want to take a measurement. Introduce the concept of while loops in Python. Lead the students through Activity 2 in the Student Book , and stop before the "Slow Down" section. Explain that the while loop is running as fast as the computer can go, but this slows down other computer processes. Additionally, it makes it hard for the students to read the values. Introduce the concept of waiting between measurements. Finish Activity 2.

For advanced students who are asking the question: *How did they know how to read from the ADC?*. Let them go or lead them through Activity 3 in the Student Book.

Depending on how advanced the students are, explain that although we have a calibration function from Volts to meters, the Python script can only read the ADC value. Therefore, we need a way to convert from the ADC value to meters. Lead the students through Activity 4 in the Student Book.

For all students, explain that the ADC value needs to be converted to a distance measurement, just like the students had to do in the introduction activity. Lead the students through Activity 5 in the Student Book to convert the ADC values to distance.

3) Ending The Lesson

Review what was accomplished today, and how it relates to the overall goal of flying the drone:

In the previous lesson, we:

1. created a function that converted Volts to meters

In this lesson, we:

1. learned that the drone needs to read and convert the measurements *fast*, and so we needed to *automate* this process.

2. learned that the Pi cannot read the analog voltage from the IR sensor, so the value needed to be converted to a digital value using the ADC

3. learned how to read and print the ADC value on the Pi.
4. learned how to convert the ADC value to meters.

If students want more practice with Python, they can go through this online Python activity on their own time.

Useful Resources and References

Python vocab:

Scripts:

This is code written in a text file and saved with ".py" at the end in order to run a lot of code at once that you can work on remotely. This is used when you want Python to preform a long chunk of code at a time or if the code spans multiple files.

Function calls:

The most simple callable object in Python (just means you put in an argument and Python retuns an object/objects). A function call specifically takes in the argument (or parameters) of a mathematic function and return values or "answers"

Printing:

example

By typingin `print()` and putting what you want the screen to show in the parentheses, Python will then print out the specified message.

Function calls definition

Subsection G.2
Middleware: ROS

This subsection contains an introduction to the Robot Operating System, which allows different programs running on and off of a robot to communicate. ROS creates a framework for sharing data between programs. In addition to learning relevant concepts, the goal of this section is to empower students with the ability to write a ROS publisher and subscriber on their own. Again, the specific use case in this subsection is for the IR sensor; however, the steps can be generalized for other programs on the drone. Learning how to write a ROS publisher and subscriber is a reusable robotics skill.

UNIT G.2.1
Intro to ROS

Student version (unknown ref duckiesky_high_school_student/sac2-ros-intro)

previous **warning** next (23 of 36) index
warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/sac2-ros-intro'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

KNOWLEDGE AND ACTIVITY GRAPH

Requires:

Hardware - Basestation

Previous Lesson - Sensing

Results:

Knowledge -

- Definition and purpose of robot middleware
- Basic ROS components and vocabulary: topic, message, publisher, subscriber, node, workspace

Skills -

- List and echo ROS topics
- Create a catkin workspace

1.1. Introduction to ROS

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

NGSS: HS - ETS1 - 2: Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.

ISTE: 1. d.: Understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

2) Assessments and Evidence of Understanding

Students will be able to undertand what ROS is and make a catkin workspace by the end of the lesson.

3) AGENDA (Brief Summary of Activities)

5 mins: Introducing ROS

45 mins: Reviewing important ROS concepts and commands

10 mins: Making a catkin workspace

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Teacher Materials

Basestation, a projector (if needed)

Classroom Set Up

Teachers can write a DO NOW on the board for students to set up their basestations.

1.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

> **Recommended:** 5 minutes

- **Hook**

Teachers use this link to:

1. Give students a sense of the structure of a software diagram.
2. Illustrate the challenges of software engineering in robotics
   - Lots of sensors and actuators, and they need to communicate with each other (even in this diagram, not many sensors and actuators but the connections are still fairly complicated)
   - Processes might live on different computers to spread the computing load
   - Many software modules need to work together (Need to manage their dependencies)

2) Main Lesson

> **Recommended:** 45 minutes

Present our solution to the challenges above: ROS (Robot Operating System).

Official Introduction of ROS

"ROS is an open-source, meta-operating system for your robot. It provides the services you would expect from an operating system, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. It also provides tools and libraries for obtaining, building, writing, and running code across multiple computers." (http://wiki.ros.org/ROS/Introduction)

In Short: ROS aims to support **code reuse** in robotics research and development.

Q: What are the benefits of reusing codes?

Answer:

```
- **Wikipedia Verision** "Modular programming is a software design tech-
nique that emphasizes separating the functionality of a program into in-
dependent, interchangeable modules, such that each contains everything
necessary to execute only one aspect of the desired functionality."
(https://en.wikipedia.org/wiki/Modular_programming)

- **In Short**: Easy to read, easy to test, easy to reuse, easy to main-
tain, etc.

To help students understand this, teachers might draw a comparison to
reading and writing a book like [this](https://www.durhampriory.ac.uk/
colour-your-own-medieval-manuscript-part-5/) versus a book like
[this](https://docs.duckietown.org/daffy/downloads/duck-
iesky_high_school/docs-duckiesky_high_school/branch/daffy-develop/doc-
duckiesky_high_school/out/index.html)
```

Overview of important ROS concepts:

ROS master: A node that every other node register in order to communicate with each other. It is created by running roscore command (Detailed: http://wiki.ros.org/Master)

ROS nodes: ROS nodes are programs that communicate with other programs via publishing and/or subscribing to ROS topics. (Detailed: http://wiki.ros.org/Nodes)

Messages: The information that is communicated between nodes. Messages are standardized, which means that certain message types have certain fields. For example, "ROS pose message" has two parameters: position and orientation. They are message types themselves, which contain their own parameters. (http://docs.ros.org/melodic/api/geometry_msgs/html/msg/Pose.html) This makes communication between programs a lot easier (link to the modular programming above). You can even create your own message types!

ROS Topics: Topics are what ROS messages are published and subscribed to. (Detailed: http://wiki.ros.org/Topics)

Publishers: Publishers are used to publish specific message types to specific topics.

Subscribers: Subscribers are used to read the messages being published to a ROS topic.

✓ Print out all of the topics running by entering "rostopic list" into a free window after running 'screen -c pi.screenrc' on your drone. Have messages that are being published to a topic printed out by navigating to an empty window in the screen and entering "rostopic echo [topic_name]". For example, if you wanted to see the data coming from the infrared sensor, you could enter "rostopic echo /pidrone/infrared"

KNOWLEDGE AND ACTIVITY GRAPH

Important ROS Commands:

roslaunch: roslaunch is a tool for easily launching multiple ROS nodes locally and remotely via SSH, as well as setting parameters on the Parameter Server. (Detailed:

http://wiki.ros.org/roslaunch)

roscd: roscd allows you to change directories using a package name, stack name, or special location. (Detailed: http://wiki.ros.org/rosbash#roscd)

rostopic: rostopic contains the rostopic command-line tool for displaying debug information about ROS Topics, including publishers, subscribers, publishing rate, and ROS Messages. (Detailed: http://wiki.ros.org/rostopic)

In your future lesson about ROS subscriber, you will create a ROS program to control your LED according to readings in of your sensor. In this case, there will be three nodes: sensor, LED, and controller. The controller node **subscribes** to the sensor node, receiving its value, determines how the LED should behave, and then **publishes** its message to the LED node.

3) Ending The Lesson

> **Recommended:** 10 minutes

✔ make a catkin workspace following this link.

Useful Resources and References

ROS overview

ROS Wiki

Unit G.2.2
Creating a ROS Publisher

Student version (unknown ref duckiesky_high_school_student/sac2-ros-publisher)

previous **warning** next (24 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/sac2-ros-publisher'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

Knowledge and activity graph

Requires:

Hardware -

- Basestation
- Build Parts 1 and 2 completed

Previous Lesson

- Sensing
- Introduction to ROS

Results:

Knowledge - Python loops

Skills -

- Create custom ROS messages
- Create ROS publishers
- Use ROS commands

2.1. Creating a ROS Publisher

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

NGSS: HS - ETS1 - 2: Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.

NGSS: HS - PS4 - 2: Evaluate questions about the advantages of using a digital transmission and storage of information.

ISTE: 5. c.: Break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving

2) Assessments and Evidence of Understanding

3) AGENDA (Brief Summary of Activities)

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Teacher Materials

Basestation, drone build

Classroom Set Up

Teachers can write a DO NOW on the board for students to open the python file that read and print sensor values written in the sensor lesson.

2.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

> **Recommended:** 5 minutes

Review the code of reading sensor values in the student book.

- Hook

Q: How should we make this information useful to other parts of the robots?

Answer: ROS!

2) Main Lesson

> **Recommended:** 30 minutes

Teacher uses this tutorial to explain how to creat ROS messages.

Two types of messages that might need more explaining:

1. **Header:** The header contains a timestamp and coordinate frame information that is commonly used in ROS
2. **Float:** More precise measurement than integers

Then, students follow the teacher step by step to create a ROS publisher, which is detailed in the student book.

3) Ending The Lesson

> **Recommended:** 15 minutes

Time to visualize our progress!

Open up a new terminal, start up roscore by typing `roscore` and hit enter in order to get nodes running.

Navigate back to the previous terminal, run the code. We can see it is still printing sensor values.

Q: What about the publisher we just created?

Answer: Open another terminal, run `rostopic list` and see the `distance` topic we just created. Then we run `rostopic info distance` to see the message type (Range) and where it is published from (infrared_node). Lastly, run `rostopic echo distance` to show the message we just created in the terminal! (It should be Range messages with data label)

Q: How can we show **our** *calibrated value (instead of the default) on the web interface?*

Answer: According to the operations manual, before taking flight, after students start the flight code by hitting `./start`, they might hit "tick 5" to go to the infrared node and hit 'Ctrl + C' to stop it. Next, they might run their own script `python ~/ws/src/pidrone_pkg/ir.py`. Now the image shown on the web interface uses the value from the publisher they just wrote!

Useful Resources and References

Python Loops

ROS Tutorial on Creating Publisher and Subscriber

Unit G.2.3
Subscriber and Open Loop

Student version (unknown ref duckiesky_high_school_student/sac2-ros-subscriber)

previous **warning** next (25 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/sac2-ros-subscriber'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

KNOWLEDGE AND ACTIVITY GRAPH

Requires:

Hardware -

- Basestation
- Build Parts 1 and 2 completed

Previous Lesson

- Sensing
- Introduction to ROS
- Creating a ROS Publisher

Results:

Knowledge

- Open loop control
- Callbacks
- Global variables

Skills -

- Create ROS subscriber
- Program a proportional controller
- Map a range of values onto another range

3.1. Subscriber and Open Loop

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

NGSS: HS - ETS1 - 2: Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.

NGSS: HS - PS4 - 2: Evaluate questions about the advantages of using a digital transmission and storage of information.

ISTE: 5. c.: Break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving

2) Assessments and Evidence of Understanding

3) AGENDA (Brief Summary of Activities)

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Materials needed

For Students:

For Teachers:

3.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

> **Recommended:** X minutes/hours

2) Main Lesson

> **Recommended:** X minutes/hours

3) Ending The Lesson

> **Recommended:** X minutes/hours

Useful Resources and References

Section H
Build: 3

Student version (unknown ref duckiesky_high_school_student/build3)

previous **warning** next (26 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/build3'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

In this phase of the build, students will be adding the essential elements of every drone– the motors, ESCs, and the flight controller. By the end of this build part, students will have a configured flight controller, and will spin the motors (without the props on). It is worthwhile to emphasize the achievement at the end of this build part: it is an exciting moment because it the first time they will see their motors spinning.

Section I
Sensors, Actuators, and Control: 3

The goal of this section is to introduce the theory behind how brushless motors work, as well as the intertial measurement unit (IMU), which is a required sensor on every drone. Understanding how the motors work motivates the need for an essential drone component: the electronic speed controllers (ESCs), which control the speeds of the motors. Understanding the IMU reveals how the drone is able to determine its own roll, pitch, and yaw, which is essential for the drone to fly.

SUBSECTION I.1
Motors

This subsection explains the theory of brushless motors and demonstrates the need for electronic speed controllers (ESCs).

UNIT I.1.1
Intro to the Motors

Student version (unknown ref duckiesky_high_school_student/sac3-motors-intro)

previous **warning** next (27 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/sac3-motors-intro'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

KNOWLEDGE AND ACTIVITY GRAPH

Hardware -

- Basestation
- Build Parts 1, 2, and 3 completed

Previous Lesson

- Drone Operation
- Sensors, Actuators, and Control: 2

Results:

Knowledge - How our motors work, what systems are in place to check them

Skills - The ability to utilize Cleanflight and a corresponding Python script interacting with Cleanflight

1.1. Introduction to Motors

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

2) Assessments and Evidence of Understanding

By the end of this lesson, students should be able to demonstrate their knowledge of the innerworkings of their motors, and should be able to run a Python script to navigate Cleanflight and operate their motors.

3) AGENDA (Brief Summary of Activities)

- 5 min: Brainstorming discussion about motors
- 10 min: Brief lecture

- 45 min: Activity with Python and Cleanflight

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

You may want to have the Python script that is already written ready to distribute to students, and to make sure all of the students have their Cleanflight flashed and working.

Materials needed

For Students: Basestation with Cleanflight and drone with Build Parts 1, 2, and 3 completed

For Teachers: Basestation with Cleanflight, drone with propellors detatched, and Python script

1.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

> **Recommended:** 5 minutes

Class discussion:

- Have the students spin their motor's manually and trace the wires back on their drone to see everything the motors are connected to
- Have them brainstorm about what each component of the FC -> ESC -> Motor chain could do
- Think back to the ethics and safety lessons...Why is smooth motor control so important?

2) Main Lesson

> **Recommended:** 10 minutes

Motor Lecture:

Motor Explanation-

There is a coil of metal within each of the 3 wires connected to the motor that allow electrons to flow through and create currents that combine to make a magnetic field that pushes and pulls on the motor to make it speed up and slow down.

ESC Explanation-

The ESCs take the input of power from the battery and digital signals from the flight controller, and then sends volts of electricity to the motor through the 3 wires.

It is the interface between the digital signal of the FC and the raw voltage going to the motors.

Accuracy Explanation=

Actuators aren't completely accurate, just like sensors aren't always completely accurate, so there is a small sensor inside the ESC that gets feedback from the motor by measuring it's back current (the voltage that comes back into the 3 wires while it's spinning). This sensor allows our drone to monitor it's motor activity mid-flight.

KNOWLEDGE AND ACTIVITY GRAPH

Vocabulary:

- *3-Phase AC*: Stands for **3-Phase Alternating Current** and describes how none of the currents from each of the 3 wires leading to the motor align at the same point, which creates the dissonance and the magnetic field.
- *Back EMF*: **Back electromotive force**, aka counter electromotive force, is the voltage that is emitted in opposition to the change in current of the motors. Each ESC has a small sensor that measures the back EMF and sends that information to the FC in order to monitor the motors as they are spinning.

3) Ending The Lesson

Recommended: 45 minutes

Showing motors in action with Cleanflight!

Note: This is the most exciting part of this lesson, so it might be best to leave extra time for it.

☀ Exercise: Use a Python script that sends a constant throttle command that spins the motors at a slow speed. Student tilts the drone to observe the FC corrections. Script prompts the student to input a roll or pitch angle. Students will observe adjacent drone motors spinning up faster. Students will move the drone to the desired angles until the motors return to equal speeds.

➡ This exercise is trying to build intuition for how the motors actuate the drone and how controllers get layered. It also gives an initial hands-on experience with a closed loop controller which they'll understand better later.

Useful Resources and References

The Actuators video of this edX lecture.

This more advanced explanation of 3-Phase AC motors.

Subsection I.2
IMU

This subsection explains the theory behind the Inertial Measurement Unit. The drone must know it’s roll, pitch, and yaw in order to fly– the IMU makes this possible.

UNIT I.2.1
Intro to the IMU

Student version (unknown ref duckiesky_high_school_student/sac3-imu-intro)

previous **warning** next (28 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/sac3-imu-intro'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

KNOWLEDGE AND ACTIVITY GRAPH

Requires:

Hardware -

- Basestation with Cleanflight
- Build Parts 1, 2, and 3 completed

Previous Lesson

- Drone Operation
- Sensors, Actuators, and Control: 2

Result:

Knowledge -

- Learn the definition and purpose of the IMU
- Be able to point out the location of the IMU on the drone
- Be able to explain how the IMU works

Skills - N/A

1.1. Introduction to IMU

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

2) Assessments and Evidence of Understanding

By the end of this lesson, students should be able to explain the basic functioning of the IMU and the laws of physics that go behind it.

3) AGENDA (Brief Summary of Activities)

- 10 min: Watch a short video.
- 20 min: Give a lecture on the physics of the IMU.
- 10 min: See the effects in Cleanflight

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Teacher Materials

Place to play a video, basestation with Cleanflight, and drone.

Classroom Setup

Teachers can write a DO NOW on the board for students to set up their basestations and Cleanflight

1.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

Recommended: 10 minutes

Watch this video

While watching, pause to emphasize the following:

1. Where the IMU is located
2. The two sensors in the IMU: Gyroscope and Accelerometer, that measure angular velocity and acceleration respectively.
3. Optional: How the Micro-Electro-Mechanical (MEM) System works - *Gyroscope*: gyros rotate -> vibration -> voltage -> ADCs *Accelerometer*: measure displacement, and then differentiate to get acceleration
4.

Note: They measure without interacting with the rest of the world! E.g. what a spacecraft would have to use. The downside of this is that it does not have long-term accuracy.

2) Main Lesson

Recommended: 20 minutes

A Smidgeon of Physics: Newton's Two Laws of Motion::

1. *Newton's First Law of Motion*: "**The vis insita, or innate force of matter**, is a power of resisting by which every body, as much as in it lies, endeavours to preserve its present state, whether it be of rest or of moving uniformly forward in a straight line."

First Law describes **Inertia**: Object's tendency to resist the change of motion.

If we want the object to move, we have to exert eternal forces, and the second law tells us the effect of this.

1. *Newton's Second Law of Motion*: F = ma (The force on an object produces acceleration)
   - F: Force on object
   - m: Mass of the object
   -

   Answer: the acceleration of the object, which is explained here
   - In short: Displacement is the change in position. Velocity is the rate of change of displacement. Acceleration is the rate of change of velocity.

The accelerometer measures the position changes and calculates all the way down to acceleration (accumulating errors along the way).

Physics Continues: Circular Motion:

Q: What causes an object to have a circular motion? Newton's Second Law tells us there must be a force exerted on the object. What is the direction of the force?

Answer: Centripetal Force! Pointing to the center of the circular path.

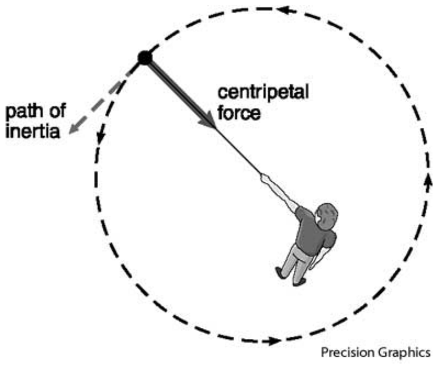

Figure 1.1. Centripetal Force Diagram

→ Example: The gravitational pull from the earth to the moon is the centripetal force causing the moon to move around the earth.

Q: For circular motion, we also care about how fast the object's orientation is changing.

Answer: Angular velocity, which measures the rate of change of the object's orientation

Teacher might refer to the first few paragraphs of this page

Similarly, angular acceleration is the rate of change of angular velocity.

The gyroscope measures rotation and calculates all the way down to angular acceleration (accumulating errors along the way).

☀ Note: The analogy between [displacement, velocity, acceleration] and [Orientation, angular velocity, angular acceleration].

✔ One student sits in a chair with eyes closed and another student spins the chair. The student in the chair guesses their orientation based on the feeling of the spin. Note that the student's accuracy declines over time, which is similar to the accelerometer and gyroscope accumulates errors.

3) Ending The Lesson

> **Recommended:** 10 minutes

Q: How do we account for the errors in the long term?

Pause for group discussion and presentation.

Answer: We need sensors that interact with the outside world, such as GPS, to know the location.

Useful Resources and References

Between times 0:21 and 5:00 of this video.

Section J
Build: 4

Student version (unknown ref duckiesky_high_school_student/build4)

previous **warning** next (29 of 36) index
warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/build4'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

In this section of the build, students will attach the camera, and finalize the drone assembly. By the end of this build part, students will be ready to fly. Be sure that students follow all of the build checkpoints to make sure that their drones are functioning properly and are ready to fly safely.

After completing all of the build parts and checkpoints, instructors can walk students (individually) through flying by following the safety checks and flight instructions in the Operations Manual.

SECTION K
Closed Loop Control

This section introduces students to closed loop control. There are many every-day examples that can be used and are suggested in the lesson plans. This section then describes one of the most popular closed-loop controllers, the proportional integral derivative (PID) controller. Although the math behind the PID involves calculus, the high-level understanding is actually quite intuitive. Additionally, when the PID is implemented on the drone, summations are used instead of integrals. For students who have not taken calculus, showing the equations is optional. However, for students who have strong math backgrounds, showing the calculus may be worthwhile to motivate continued education in mathematics.

Subsection K.1 PID Controller

A PID (proportional, integral, derivative) controller is a control algorithm extensively used in industrial control systems to generate a control signal based on error. The error is calculated by the difference between a desired setpoint value, and a measured process variable. The goal of the controller is to minimize this error by applying a correction to the system through adjustment of a control variable.

UNIT K.1.1

Intro to PID

Student version (unknown ref duckiesky_high_school_student/loop-pid-intro)

previous **warning** next (30 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/loop-pid-intro'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

KNOWLEDGE AND ACTIVITY GRAPH

Requires:

Hardware - Their completed drone and their basestation **Previous lesson** - Intro to the Motors

Result:

Knowledge -

- Students will be able to understand feedback control systems and be able to use control system vocabulary terms such as:
- Setpoint
- Control variable
- Error

Skills - N/A

1.1. Intro to PID

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

ISTE: 1. d.: Understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

2) Assessments and Evidence of Understanding

By the end of this lesson, students will understand the different between open and closed loop systems, feedback control systems, what the PID controller is used for in the case of their drones, and be able to use vocabulary terms pertaining to feedback loops and systems.

3) AGENDA (Brief Summary of Activities)

5 min: Introduction to the lesson and identify the 1D hovering problem of a drone.

40 min: Main lesson which includes learning about feedback loops, an example of feedback loops, and defining terms associated with feedback loops.

5 min: Conclusion and summary of the lesson.

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Materials needed

Teacher Materials

- Projector for displaying videos or slides
- Reference the student textbook, EdX and Youtube videos (both linked in the useful resources section below) for more detailed explanations of concept snad examples

Classroom Set Up - Allow space for students to fly their drones

1.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

✔ Exercise: Students can try manipulating their LEDs on the drone as they have done before in the build.

Hook:

- Students will already have the ability from previous parts of the build to manipulate their LEDs. To manipulate their LEDs, such as turning them on and off, is a relatively simple task: students can give it a command to tell it to turn on or off. However, there are other tasks that are not as simple. The 1D hovering problem of a drone (getting the drone to hover at a specific altitude) cannot be achieved by just sending one speed to the motors due to factors such as the actions of the actuators and noise of the environment.
- Introduce the idea of feedback loops.

2) Main Lesson

- Define open loop vs closed loop systems

✔ Exercise: Do this in two stages. First to demonstrate an "open-loop", a student looks at a target they're supposed to go to, gets blindfolded, and tries to walk there. To demonstrate a "close-loop system", do the same thing with a partner and without being blindfolded, and get the partner to yell directions.

- These two exercises would have shown the difference between open and closed loops: closed loop systems would require constant checking of the distance remaining between their current location and the goal location.

✔ Exercise: Teachers draw a block diagram of an closed-loop controller. Show the single input into the controller: the setpoint. Then draw the feedback loop and explain that there are now two inputs to the controller: the setpoint and the error (how far the robot is from the setpoint). Let students label or identify each part of the diagram.

- Emphasize that closed-loop systems works in iterations, based on the current setpoint and measurements it determines an output, then does the same process over again after some delay.
- Explain important terms and definitions involved with the PID (proportional, integral, derivative) controller

✔ Exercise: Students can brainstorm some of examples of feedback loops.

- Choose an example of a feedback loop and explain what each of the terms would be represented by in the example:
- ie: body thermoregulation, reflexes, hormone feedback loops, heart and blood pressure in the body
- Define the error term
- Explain the general reason for the PID controllers in systems

3) Ending The Lesson

✔ Exercise: Students try flying their drone, and try identifying what the process variable, setpoint, and control variable would be for the 1D drone hovering problem.

- Go over what the three terms are representing in the 1D drone hovering problem.
- Feedback loops are used in many technological, engineering, system applications.

Useful Resources and References

- Teachers can see EdX lectures on PID Control

Here is a helpful video explaining PID controllers

UNIT K.1.2
The Three PID Terms

Student version (unknown ref duckiesky_high_school_student/loop-pid-terms)

previous **warning** next (31 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/loop-pid-terms'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

KNOWLEDGE AND ACTIVITY GRAPH

Requires:

Hardware

- Basestation
- Part 1 of drone

Previous lesson - Intro to PID

Knowledge -

- What each term of a PID controller does
- Why each term is important
- Rough mathematical understanding of how each term works

Skills - Tuning a PID loop

2.1. The Three PID Terms

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

ISTE: 1. d.: Understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

2) Assessments and Evidence of Understanding

By the end of this lesson, students will understand the different terms used in PID systems.

3) AGENDA (Brief Summary of Activities)

5 min: Introduction to lesson

40 min: Go over definitions of PID terms and their effects

5 min: Conclusion

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Materials needed

Teacher Materials

- Projector for displaying videos or slides
- Reference the student textbook, EdX and Youtube videos (both linked in the useful resources section below) for more detailed explanations of concept snad examples

2.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

- Hook: They will use the vocabulary and concepts that they learned in the last lesson to be able to understand how PID works, the terms associated with PID, and its effects in the 1D Hovering Drone Problem.

2) Main Lesson

- Show students video explaining PID controllers

The Proportional Term

✔ Exercise: Students try different extreme values for the P term in the simulation

- Explain the P term and try tuning in simulation.
- Explain the effects of having a higher or lower proportional gain constant.
- Explain affects of the proportional term regarding the altitude problem.

The Derivative Term

✔ Exercise: Students try different extreme values for the D term in the simulation.

- Explain the D term and try tuning in simulation.
- Explain affects of the derivative term regarding the altitude problem and its limitations.

☀ Teachers can show graphs of different levels of damping: over, under, critical.

The Integral Term

✔ Exercise: Students try different extreme values for the I term in the simulation

- Teachers will explain the I term
- Explain affects of the integral term regarding the altitude problem and its limitations.

The overall control function

- Explain the overall control function

Show diagram of PID controller Block Diagram:

The figure below summarizes the inclusion of a PID controller within a basic control loop.

Figure 2.1. PID Controller Block Diagram

Tuning:

- Explain why tuning a PID controller is necessary

Effects of K_p::

- Describe the effects of K_p

Effects of K_i:

- Describe the effects of K_i

Effects of K_d:

- Describe the effects of K_d

3) Ending The Lesson

- Explain that while we are looking at the terms from the context of 1D altitute control, they are also used for others such as angle, velocity, position, etc of the drone.
- Summarize the effects and what each term is

Figure 2.2. General Effects of Control Terms

Useful Resources and References

- Teachers can see EdX lectures on PID Control

Unit K.1.3

Implementation of PID

Student version (unknown ref duckiesky_high_school_student/loop-pid-implementation)

previous **warning** next (32 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/loop-pid-implementation'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

Knowledge and activity graph

Requires:

Hardware

- Basestation
- Drone
- Build Part 1 Completed

Previous lesson - The Three PID Terms

Results:

Knowledge - Python classes

Skills - Learn how to implement a 1D PID controller

3.1. Implementation of PID

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

ISTE: 1. d.: Understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

2) Assessments and Evidence of Understanding

By the end of this assignment, students will be able to implement a 1D PID Controller.

3) AGENDA (Brief Summary of Activities)

10 min: teach what a python class is for

40 min: Implement the 1D controller

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Materials needed

Teacher Materials

- Projector for displaying videos or slides
- Look under resources section for more information about coding and classes

3.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

- Explain what classes in python allow you to do
  - allows to have multiple copies of code running at once

2) Main Lesson

- Students will inmplement a PID controller for a simulated drone that can only move in 1D, the vertical dimension. You can control the speed the motors spin on the drone, which sets the thrust being generated by the propellers.

  ☀ they will do this with class tenplate code and stadalone simluator for testing

  ☀ In this system, the process variable is the drone's altitude, the setpoint is the desired altitude, and the error is the distance in meters between the setpoint and the drone's altitude.

3.3. Problem 1: Implement an Idealized PID

Exercises

1. Implement the step method to return the constant K. At what value of K does the drone takeoff? What could happen if K were set too high on a real drone? Set K to 1300 for the remainder of the questions.
2. Implement the P term. What happens when the absolute value of K_p is very large? What happens when its absolute value is very small? Can you tune the P term to stop oscillations? Why or why not?
3. Implement the D term. Set K_p to zero. What happens when K_d is 50? 500? 5000?
4. Now tune K_p and K_d so that the drone comes to a steady hover. Describe the tradeoff as you change the ratio of K_p to K_d. Can the drone stabilize at its target (zero steady-state error)? Why or why not?
5. Implement the I term and observe the difference between PD and PID control. What role does the I term play in this system? What happens when K_p and K_d are set to zero?
6. Implement the reset method and test its behavior. If implemented incorrectly, what

problems can you anticipate reset causing?

7. Finally, tune the constants in your PID controller to the best of your abilities. When the setpoint is moving, the drone should chase the setpoint very closely. When the setpoint is still, the drone should converge exactly at the setpoint and not oscillate. Report your tuning values.

3.4. Problem 2: Tuning a PID with Latency

Now, we introduce latency! Run the simulation as `python sim.py -l 6` to introduce 24 milliseconds of latency (six steps of latency running at 25 hz).

Exercises

1. Tune the constants in your PID controller to the best of your abilities. The drone should chase the setpoint very closely, but will converge more slowly when the setpoint is still. Report your tuning values.
2. Compare your tuning values to the values you obtained in problem 1.
3. Explain the effect of latency on each control term.

3.5. Problem 3: Tuning a PID with Latency, Noise, and Drag

In the most realistic mode, you will tune a controller with latency, noise, and a drag coefficient. You can do this with the command line arguments `python sim.py -l 3 -n 0.5 -d 0.02` to be most realistic to real-world flight.

Exercises

1. Tune with these arguments to be as good as possible. Report your tuning values.
2. Compare your tuning values to the values from problems 1 and 2.

Run `python sim.py -h` to see the other simulator parameters. We encourage you to experiment with those and observe their effects on your controller.

1) Tuning 1D Controls: Part 1: Planar Tuning

In this portion of the project, you will be tuning the low rate integral terms of the PID controllers that we've provided.

3.6. Trimming your Drone

Due to differences in the weight distribution and other factors that cause asymmetries, the drone will tend to initially drift in a particular direction. In order to tune your altitude PID, the planar motion of the drone needs to be controlled. This is important so that the drone does not fly uncontrollably across the room while you're trying to tune its altitude controller. To control the drone's planar motion while you're tuning the altitude, we've created and tuned PIDs to do this for you, but you will need to tune the initial low-rate integral terms to account for the uneven weight distribution specific to your drone. You will first use the provided altitude PID to tune the planar controllers, and then you will tune your altitude PID with the tuned planar controllers.

3.7. Problem 1: Understanding the Controller

Our controller is a dual I-term (integral term) PID controller. The high-rate I-term changes quickly, allowing fast response to changes. The low-rate I-term changes slowly, allowing the drone to adjust to systemic sources of error. The provided PID gains have been pretuned to this drone hardware, and should not need significant modification for your specific drone. But, the initial low I-terms do need to be adjusted based on the static error of your specific drone.

Exercises

1. Name a source of static error that the low-rate I term can correct for.
2. Name two sources of dynamic error that the high-rate I term can correct for.

3.8. Problem 2: Tune the Throttle

The first step in the tuning process is finding an initial throttle value that allows your drone to have a smooth and controlled takeoff. To do this, you'll be adjusting the value of `throttle_low.init_i` in *pid_class.py*. This is the initial value of the low-rate (slow changing) integral term for the throttle, which controls altitude. The default value is 100. you will tune this value by having the drone take off, observing its behavior, and modifying the value accordingly. Each time you wish to change the value, you will need to restart *pid_controller.py* to use the new value.

Setup

1. Prepare your drone to fly over a highly textured planar surface1.
2. Navigate to `4 of the screen.
3. Quit the program by pressing ctrl-c.

Exercises

1. In this screen (`4), use a text editor (such as vim or nano) to modify `throttle_low.init_i` in *pid_class.py* to test out different values for `throttle_low.init_i`. Be cautious when modifying this value because the drone could take off abruptly with a value that is too high. The specific `throttle_low.init_i` value is drone specific, but typical values range between 50 and 150. Try both of these values and two more values between then. In one sentence, describe the drone's behavior as a result of changing the value up and down.
2. Now find the value for which your drone is able to have a smooth and controlled takeoff. The goal is to reduce the overshoot and undershoot for the drone to takeoff and fly stable at 0.3m. Try changing this value in increments of 10 and then 5 until you find a value that allows the drone to take off at a reasonable rate. Record this value in your answers.

3.9. Problem 3: Set the Trim

Next you will set the trim on roll and pitch. You will do this by tuning the low I-terms to adjust for the static errors that exist on your drone. The default value is 0, and positive values will move the drone to the right or forward, and negative to the left or backward, depending on the axis you're modifying. Note that you may need to repeat this process periodically, for example after a crash or the like. When performing this process, each time make sure that you:

- Place the battery in the same place each time as much as possible so the weight is distributed the same.
- Plug the flight controller while the struts are fully engaged and the drone is level, so the gyros are well calibrated.
- Always place the drone so that the camera is closer to you and the skyline is farther away.

Setup
Modify *pid_controller.py* to print out the low rate integral terms of the PIDs by finding the block of code shown below and uncommenting the following print statements

```
print 'roll_low.init_i', pid_controller.pid.roll_low.init_i
print 'pitch_low.init_i', pid_controller.pid.pitch_low.init_i
```

You will also need to set the `verbose` variable in this file to zero so that these print statements will not be overridden by the other print statements: `verbose = 0`

While flying, the low-rate I-terms will change to account for the static flight error, and when you disarm the drone, the initial low-rate I terms will be set to these changed values, thus allowing the low-rate I terms to start at this corrected value. Eventually, these values will converge, and your drone will no longer drift. Once converged, you will save the values by modifying the variables `self.roll_low.init_i` and `self.pitch_low.init_i` in `pid_class.py` to the corresponding value printed in `4 of the screen after disarming. This will store the initial low-rate I-terms between flights.

Exercises

1. Perform one flight. After the drone takes off, do not give it movement commands but allow it to drift.
2. Disarm the drone before it flies too far in any direction.
3. Write down the low-I values printed in `4 of the screen.
4. Pick up and move the drone by hand back to the center of the flying area.
5. Repeat steps 1-4 until the values that are printed out after disarming have converged (roughly when the change in magnitude is less than 1).
6. Once these values have converged, record these values in your answers.

Footnotes
1A flat posterboard scribbled or written on with marker will work.

1) Tuning 1D Controls: Part 2: Altitude Tuning

In this part, you will be transferring the altitude PID you created in part 1 onto your drone. You will then tune the PID gains on your drone as you did in the simulator.

3.10. Problem 1: Flying with Your Altitude PID!

Now that the planar PIDs are tuned, and you have found a value for `throttle_low.init_i` that allows the drone to take off at a reasonable rate, you will be using your altitude PID to control the height of the drone. To tune your altitude PID, you will first use the Ziegler-Nichols tuning method to generate an initial set of tuning parameters. You will then fine tune these parameters similar to how you tuned the drone in simulation.

To use your PID, you'll be running *student_pid_controller.py* instead of *pid_controller.py*. This will allow your PID to run alongside our planar PIDs, and on top of our throttle low-rate I-term which you found previously. Your PID will be responsible for keeping the drone flying steady vertically.

Setup

Change directories to `~/ws/src`. Run `git clone https://github.com/h2r/project-pid-yourGithubName.git`. In your repo, change "pidrone_project3_pid" to "project-pid-yourGithubName" in *package.xml* and "project(pidrone_project3_pid)" to "project(project-pid-yourGithubName)" in *CMakeLists.txt*. Also remove the msg folder, and comment out "add_message_files" in *CMakeLists.txt*. Then change directories back to `~/ws/` and run `catkin_make --pkg project-pid-yourGitHubName`.

OR

Use the `scp` command to transfer *student_pid_class.py*, *student_pid_controller.py*, and *z_pid.yaml* from the repo on your base station to the scripts folder of your drone (`~/ws/src/pidrone_pkg/scripts/`). In the instructions below, instead of using `rosrun`, you may use `python` to execute your scripts.

Change directories into `~/ws/src/pidrone_pkg` and modify *pi.screenrc* to start up with your altitude pid by changing `python pid_controller.py\n` to `rosrun project-pid-yourGitHubName student_pid_controller.py\n`. Prepare your drone to fly and then navigate to `4 of the screen. Press ctrl-c to quit student_pid_controller.

In this screen (`4), modify `~/ws/src/project-pid-yourGitHubName/z_pid.yaml` by setting $K$ to 1250 and the rest of the gain constants to 0. Now run `rosrun project-pid-yourGitHubName student_pid_controller.py` to fly with your altitude PID.

Exercises

Fly your drone and observe its flight. Tune K_p by slowly increasing its value between flights until you can see the drone moving up and down with uniform oscillations. Each time you will need to quit the controller, edit `~/ws/src/project-pid-yourGitHubName/z_pid.yaml`, and then run `rosrun project-pid-yourGitHubName student_pid_controller.py` again to use the new PID gains.

1. Record your final K_p value that causes uniform oscillations as K_u, the ultimate gain.

2. Fly your drone and pause the altitude graph on the web interface when you see two peaks. Find the time difference between these two peaks and record this value as T_u, the ultimate period.

3. Use your K_u and T_u values to compute K_p, K_i, and K_d. Refer to the equations in the Ziegler-Nichols section in the introduction to this project. Record these values and change *z_pid.yaml* accordingly.

4. Fly your drone with the set of tuning values generated by the Ziegler-Nichols method. Note that the Ziegler-Nichols method should enable safe flight, but will probably not control your drone's altitude very well! Empirically tune the gain constants in *z_pid.yaml* on your drone as you did in the simulator portion of this project. 2 Record

your final tuning values.

Take a video of your drone flying first using our altitude pid by running *pid_controller.py* in `4, then take a video of your tuned pid by running *student_pid_controller.py* in `4. See if you can get yours to track the altitude setpoint better than ours! The drone should get to the setpoint quickly and stay there without bouncing up and down. *Submit these videos in Github Classroom as 'original_controller' and 'student_controller'*

Footnotes

2 Use the graph on the web interface to observe the drone's behavior as it oscillates around the 0.3m setpoint the drone's ability to hover at the setpoint. When observing the drone itself, try to get eye-level with the drone to just focus on the the altitude and ignore the planar motion; it is easier to focus on one axis at a time when tuning the PIDs. The planar axes can be re-tuned after you tune your altitude pid if need be.

1) Tuning 1D Controls: Part 3: Position Control

Thus far, the planar PIDs have been used to control the velocity of the drone; now, you will use cascaded PIDs to control the position of the drone. The cascaded PIDs are set up so that the position controller forms the outer loop which uses the position error to provide setpoint velocities for the inner loop velocity controller.

2) How to Engage Position Control

Engaging position control involves two steps. First you have to tell the drone to "remember" a frame. You can do this using the *r* key. This will save the frame which the drone will attempt to fly directly above. Next you have to engage position control. You can engage this mode with the *p* key, and disengage with *v* for velocity control. So the procedure is to first save a frame (target location for position hold) using *r* and then shortly after (before drifting too much) type *p*.

Note: Position hold works best over a textured surface with lots of visual contrast. Even when doing position hold, always be ready to kill in case of a mishap. Especially be careful when looking at other windows.

3) Position Control Demo

This video demonstrates the drone doing a zero velocity hover and drifting in the scene. Then we turn on position hold (you can tell when it is engaged when the drone's throttle drops) and it holds its position for several minutes.

Then we turn off the position hold so you can see it drift again, and then turn it on again at the end and land. You can tell when it is turned on because we move the drone back to the center of the flight area before each hold.

3.11. Problem 1: Flying with Velocity Control

First, you are going to experiment with flying your drone in velocity control and controlling its motion with the keyboard keys. Based on observations and knowledge of the controllers, you will then explain the inner workings of the velocity PIDs in your own words.

Setup
Prepare your drone to fly over a highly textured planar surface. Make sure there is space for the drone to fly around.
Exercise
Fly your drone in velocity control (the default control) and make sure there is room to fly to the right. Press and hold 'L' and observe the drone's motion, and release 'L' to stop the drone from moving.

1. Explain what the following key terms are in this controller, and how they change to cause the drone to move when you press 'L' and stop when you release: setpoint, error, control variable, process variable, proportional term, integral term, derivative term. We are looking only for a higher level description to demonstrate understanding of the PID controllers.

2. Try flying in velocity mode over a blank white poster board. Be careful! What do you notice about the drone's behavior, and what do you suspect causes this?

3.12. Problem 2: Flying with Position Control

Now you are going to fly your drone in position control and experiment with controlling its motion with the keyboard keys. Based on observations and knowledge of the controllers, you will then explain the inner workings of the position PIDs in your own words.
Setup
Prepare your drone to fly over a highly textured planar surface. Make sure there is space for the drone to fly around.
Exercises
1. Engage position hold using the procedure described above. Observe the drone's behavior. How is it different from just velocity control?
2. How long are you able to hold position? Ideally you should be able to do this in one spot for an entire battery. If not, try re-tuning your I-term preloads above. If you're flying on the power supply instead of a battery, the drone should stay in place indefinitely, but you can stop it after 5 minutes.
3. While flying in position control, make sure there is room for the drone to fly to the right and then take note of the desired position in `4 of the screen. Now press the 'L' key in the user interface and note the new desired x-position of the drone; it should be 0.1m to the right of the drone's last position. Explain what the following key terms are in the outer loop position controller, and how they change to cause the drone to move and stop 0.1m to the right after you press 'L' in position control: setpoint, error, control variable, process variable, proportional term, integral term, derivative term. We are looking only for a higher level description to demonstrate understanding of cascaded PID controllers.
4. Try flying in position control over a uniform surface such as the floor in 121, or unpatterned carpet. Echo the state of the drone by typing `rostopic echo /pidrone/state` into an empty window in the screen. Note the position data, and explain your observations of how well the drone is able to estimate its position. How long is it able to hold position? Does the drone move correctly when you use the arrow keys?

Take a one minute video of your drone flying in velocity control, and then engage position control.

1) Ending The Lesson

- Students now should be able to tune their PID loops!

Useful Resources and References

- Teachers can see EdX lectures on PID Control
- Basics of python here

Section L
Localization

Localization allows the drone to know where it is above a predefined map. In the case of the drone, the “map” is a picture of the surface that the drone is flying over. The goal of this section is for students to understand the high-level concepts of localization using fun exercises, and then to use the localization features on the drone to fly to specific positions in maps that they create.

Subsection L.1
Camera

This section describes how the downward-facing camera on the drone is used to determine the x and y positions of the drone flying over a map. Additionally, this section describes how the camera uses the change in consecutive images to determine the x and y velocities of the drone over any textured surface.

Unit L.1.1
Comparing Images

Student version (unknown ref duckiesky_high_school_student/localization-camera-images)

previous **warning** next (33 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/localization-camera-images'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

Knowledge and activity graph

Requires:

Hardware

- Basestation
- Completed Build Part 4

Previous lesson - N/A

Results:

Knowledge - How the camera allows the drone to measure the drones planar position and velocity

Skills - N/A

1.1. Comparing Images

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

2) Assessments and Evidence of Understanding

By the end of this lesson, students should be able to ...

3) AGENDA (Brief Summary of Activities)

X min:

X min:

X min:

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Teacher Materials:

Classroom Setup:

Teachers can write a DO NOW on the board for students to ...

1.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

> **Recommended:** X minutes

Hook:

2) Main Lesson

> **Recommended:** X minutes

✔ Exercise:

3) Ending The Lesson

> **Recommended:** X minutes

Useful Resources and References

1.

UNIT L.1.2
Drone Localization

Student version (unknown ref duckiesky_high_school_student/localization-camera-localization)

previous **warning** next (34 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/localization-camera-localiza-
tion'.
```

Location not known more precisely.

Created by function `n/a` in module `n/a`.

KNOWLEDGE AND ACTIVITY GRAPH

Requires: test to see changes in ops manual

Hardware -

- Basestation
- Completed Build Part 4

Previous lesson - Comparing Images

Results:

Knowledge - Definition and purpose of localization

Skills

- Make a map for the drone to localize over
- Move the drone to global positions within the map

2.1. Drone Localization

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

2) Assessments and Evidence of Understanding

By the end of this lesson, students should be able to ...

3) AGENDA (Brief Summary of Activities)

X min:

X min:

X min:

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Teacher Materials:

Classroom Setup:

Teachers can write a DO NOW on the board for students to ...

2.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

Recommended: 10 -15 minutes

Hook:

In autonomous mobile robotics estimate of position is crucial. If you covered your eyes while standing in a room your other four senses in your body would help you to survey your surroundings. Robots how ever do not pocess senses like humans but we can simulate them with sensors and cameras. For example a self driving car has sensors and mounted cameras as an essential component that collect data to let the computer know a rough estimation of the machines surroundings, like if another car were infront of it. This process is also known as **localization**.

2) Main Lesson

Recommended: 45 minutes

In order for the drone to use localization you will be using two algorithms on the PiDrone **Monte Carlo** and **Fast S.L.A.M.**. These two algorithms are desgined to target specific drone flying scanrios; scenario one (Monte Carlo) the drone is provided with a map like the one implemented into this lesson, and scenario two where the drone would use its sensors to simultaniously develope a map of its enviroment (Fast Slam).

For the purpose of this course we will be using the Monte Carlo localization (M.C.localization) which refers to the priniciple of using random sampling to model a complicated unavoidable process. In a more in depth analysis the M.C. localization is a very popular particle filter algorithm. In every new frame captured by the drones camera the filter will apply a motion prediction to adjust each possible position of the drone. At each movement correction the motion predictions will be resampled until it finds a match.**Do not worry if you still feel uneasy on the concept, we recomend doing the localization activities for better understanding**

✔ Exercise: if a jigsaw puzzle is available, otherwise with pictures of one, give students a few pieces and let them try to figure out where their pieces come from by looking at the finished puzzle picture on the box.

The drones primary sensor is the pi camera facing downward. To complete understanding of how the particle filters allow localization we will take a closer look at the process the camera extracts features and shapes to determine its position. To process information from the camera we will use a open source computer vision library callled Open-

VC. From a computers perspective a **feature** is a point of interest. Unlike humans that are able to identify a point of interest on a picture like; texture, patterns, color, or what ever may call our attention, a computer needs a precise definition of the point of interest. Features can also be defined as areas in an image where the pixel intensities change rapidly. Once a feauture is extracted the OpenCV will give us a keypoint and descriptor for each feature with its corresponding (x,y) coordinates.

3) Getting Set Up

1. place your directory in the /ws/src folder on your drone.
2. you should find "package.xml" and "CMakeLists.txt" which you need to modify your package.
3. On line 3 of "package.xml" you need to replace the default github name with your github name so it matches the nam in your directory. Do the same on line 2 of "CMakeLists.txt".
4. navigate to the /ws folder and run the following line of code.

catkin –pkg project-localization-slam-2019-yourGithubName

(this allows your package to be ros-runnable from the pidrone_pkg. This is a one time step)

4) Using the Localization

Coming Soon

5) Vocabulary

Localization: the fact of being or becoming located or fixed in a particular place.

S.L.A.M. (acronym): simultaneous localization and mapping problem

6) Ending The Lesson

Recommended: 10 - 15 minutes

✓ Exercise: Have the drone use it to localize and fly to different targets.

Useful Resources and References

1. Oxford Dictionary - Definition of localization

Section M
Materials

This section contains other supplementary materials for the teacher textbook.

Glossary in Student Book (unknown ref duckiesky_high_school_student/materials-glossary)

previous **warning** next (35 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/materials-glossary'.
```

Location not known more precisely.
Created by function `n/a` in module `n/a`.

Bibliography in Student Book (unknown ref duckiesky_high_school_student/bibliography)

previous **warning** (36 of 36) index

warning

```
I will ignore this because it is an external link.

 > I do not know what is indicated by the link '#duck-
iesky_high_school_student/bibliography'.
```

Location not known more precisely.
Created by function `n/a` in module `n/a`.

UNIT M.0.3
Teacher Template

KNOWLEDGE AND ACTIVITY GRAPH

Requires:

Hardware -
Previous lesson -

Results:

Knowledge -

-

Skills -

-

3.1. Lesson Title

1) STANDARDS: Next Generation Science Standards (NGSS) and International Society for Technology in Education (ISTE)

2) Assessments and Evidence of Understanding

By the end of this lesson, students should be able to ...

3) AGENDA (Brief Summary of Activities)

X min:

X min:

X min:

4) Differentiation *(strategies for grouping, ELL, and inclusion)*

5) Advanced preparation/Materials/Set Up (Including Misconceptions)

Teacher Materials:

Classroom Setup:

Teachers can write a DO NOW on the board for students to ...

3.2. SCRIPT OF TEACHING AND LEARNING ACTIVITIES

1) Introducing The Lesson

Recommended: X minutes

Hook:

2) Main Lesson

> **Recommended:** X minutes

✔ Exercise:

3) Ending The Lesson

> **Recommended:** X minutes

Useful Resources and References

1.

Section N Bibliography

Please see [1].

[1] Roger R Labbe Jr. Kalman and bayesian filters in python. Published as a Jupyter Notebook hosted on GitHub at https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python and in PDF form at https://drive.google.com/file/d/0By_SW19c1BfhSVFzNHc0SjduNzg/view?usp=sharing (accessed August 29, 2018), May 2018.