
Barnabas-Bot

Level 1 - Introduction to Robotics

OPEN-SOURCE CURRICULUM

[Revision 1.00]



REVISION HISTORY

Creation Date: 1/1/2017
Modified Date: 12/19/2018
Author(s): Edward Li, Eric Lin
Contributor(s): Andrew Pilcher, Victoria Lin

Revision History:

- 1.00: First Revision
- 1.01: Final updates before first release

LICENSE



Attribution-ShareAlike

CC BY-SA

This license lets others remix, tweak, and build upon your work even for commercial purposes, as long as they credit you and license their new creations under the identical terms. This license is often compared to “copyleft” free and open source software licenses. All new works based on yours will carry the same license, so any derivatives will also allow commercial use.

[View License Deed](#) | [View Legal Code](#)

TABLE OF CONTENTS

REVISION HISTORY	1
LICENSE	2
TABLE OF CONTENTS	3
COURSE OVERVIEW	18
Course Description	18
Prerequisites	18
Background	18
Disciplinary Core Ideas	18
FULL COURSE MATERIALS LIST	19
WEEK 1 OF 16: INTRODUCTION TO ROBOTICS	20
Lesson Overview	20
Disciplinary Core Ideas	20
Learning Target(s)	20
Technical Skills	20
Life Skills	20
Essential Questions	20
Key Vocabulary	20
Additional Resources	20
Videos	20
Documents	20
Other	20
Depth of Knowledge Levels Addressed	20
Barriers To Learning	20
Anticipatory Set	20
Independent Practice	21
Final Assessment, Project, or Product	21
Lesson Materials	21
WEEK 1 OF 16: INTRODUCTION TO ROBOTICS	22
Lesson Plan	22
Lesson Plan	22
Instructor Mindset	22
Step 1: Introducing Barnabas Robotics (10 minutes)	22

Project Objective	22
Creating a Culture of Encouragement	22
Step 2: Connecting the Disciplines (40 minutes)	23
Background Knowledge	23
Discussion	24
WEEK 1 OF 16: INTRODUCTION TO ROBOTICS	26
Reflection	26
Comprehension	26
Challenges	26
Enjoyment	26
Mindset	26
Community	26
Supporting Materials	26
WEEK 2 OF 16: TEAMWORK THROUGH ROBOTICS	28
Lesson Overview	28
Disciplinary Core Ideas	28
Learning Target(s)	28
Technical Skills	28
Life Skills	28
Essential Questions	28
Key Vocabulary	28
Additional Resources	29
Videos	29
Documents	29
Other	29
Depth of Knowledge Levels Addressed	29
Barriers To Learning	29
Anticipatory Set	29
Independent Practice	29
Final Assessment, Project or Product	29
Lesson Materials	29
WEEK 2 OF 16: TEAMWORK THROUGH ROBOTICS	31
Lesson Plan	31
Review	31
The Trash-Bot Challenge (45 minutes)	31
Project Overview	31
Vocabulary	31

Step 1: Forming Teams (10 minutes)	31
Step 2: The Design Process (10 minutes)	32
Step 3: The Build Process (40 minutes)	32
WEEK 2 OF 16: TEAMWORK THROUGH ROBOTICS	33
Reflection	33
Comprehension	33
Challenges	33
Enjoyment	33
Mindset	33
Community	33
WEEK 3 OF 16: COMPUTER-AIDED DESIGN	34
Lesson Overview	34
Disciplinary Core Ideas	34
Learning Target(s)	34
Technical Skills	34
Life Skills	34
Essential Questions	34
Key Vocabulary	34
Additional Resources	34
Videos	34
Documents	35
Other	35
Depth of Knowledge Levels Addressed	35
Barriers To Learning	35
Anticipatory Set	35
Independent Practice	35
Final Assessment, Project, or Product	35
Lesson Materials	35
WEEK 3 OF 16: COMPUTER-AIDED DESIGN	36
Lesson Plan	36
Review	36
Step 1: What Is CAD? (15 minutes)	36
Step 2: CAD Walkthrough (45 minutes)	36
Preemptive Vocabulary	36
OnShape Basics	39
Writing on Your Robot's Body	39
Putting Your Robot Together	41
Vocabulary	45

WEEK 3 OF 16: COMPUTER-AIDED DESIGN	47
Reflection	47
Comprehension	47
Challenges	47
Enjoyment	47
Mindset	47
Community	47
WEEK 4 OF 16: ADVANCED C.A.D.	47
Lesson Overview	48
Disciplinary Core Ideas	48
Learning Target(s)	48
Technical Skills	48
Life Skills	48
Essential Questions	48
Key Vocabulary	48
Additional Resources	48
Videos	48
Documents	48
Other	49
Depth of Knowledge Levels Addressed	49
Barriers To Learning	49
Anticipatory Set	49
Independent Practice	49
Final Assessment, Project, or Product	49
Lesson Materials	49
WEEK 4 OF 16: ADVANCED C.A.D.	50
Lesson Plan	50
Review	50
Step 1: Creating An Assembly File (10 -15 minutes)	50
Share your robot with us!	52
WEEK 4 OF 16: ADVANCED C.A.D.	53
Reflection	53
Comprehension	53
Challenges	53
Enjoyment	53
Mindset	53
Community	53

WEEK 5 OF 16: 3-D PRINTING	53
Lesson Overview	54
Disciplinary Core Ideas	54
Learning Target(s)	54
Technical Skills	54
Life Skills	54
Essential Questions	54
Key Vocabulary	54
Additional Resources	54
Videos	54
Documents	54
Other	55
Depth of Knowledge Levels Addressed	55
Barriers To Learning	55
Anticipatory Set	55
Independent Practice	55
Final Assessment, Project, or Product	55
Lesson Materials	55
WEEK 5 OF 16: 3-D PRINTING	56
Lesson Plan	56
Review	56
Step 1: Introduction To 3-D Printing (25 minutes)	56
Vocabulary	57
Step 2: Industrial Design (20 minutes)	58
WEEK 5 OF 16: 3-D PRINTING	59
Reflection	59
Comprehension	59
Challenges	59
Enjoyment	59
Mindset	59
Community	59
WEEK 6 OF 16: TIME TO DECORATE	59
Lesson Overview	60
Disciplinary Core Ideas	60
Learning Target(s)	0
Technical Skills	60
Life Skills	60

Essential Questions	60
Key Vocabulary	60
Additional Resources	60
Videos	60
Documents	60
Other	60
Depth of Knowledge Levels Addressed	60
Barriers To Learning	60
Anticipatory Set	60
Independent Practice	60
Final Assessment, Project, or Product	61
Lesson Materials	61
WEEK 6 OF 16: TIME TO DECORATE	62
Lesson Plan	62
Step 1: Decorating The Robot (60 minutes)	62
WEEK 6 OF 16: TIME TO DECORATE	63
Reflection	63
Comprehension	63
Challenges	63
Enjoyment	63
Mindset	63
Community	63
WEEK 7 OF 16: INTRODUCTION TO CIRCUITS	64
Lesson Overview	64
Disciplinary Core Ideas	64
Learning Target(s)	64
Technical Skills	64
Life Skills	64
Essential Questions	64
Key Vocabulary	64
Additional Resources	65
Videos	65
Documents	65
Other	65
Depth of Knowledge Levels Addressed	65
Barriers To Learning	65
Anticipatory Set	65
Independent Practice	65

Final Assessment, Project, or Product	65
Lesson Materials (per student)	65
WEEK 7 OF 16: INTRODUCTION TO CIRCUITS	66
Lesson Plan	66
Review	66
Step 1: Electronic Basics (20 minutes)	66
Vocabulary	66
Step 2: Building Our First LED Circuit (1.5V) (10 minutes)	67
Vocabulary	68
Step 3: Building The 3 Volt Circuit (15 minutes)	69
Step 4: The Two LED Circuit (15 minutes or as time permits)	72
WEEK 7 OF 16: INTRODUCTION TO CIRCUITS	73
Reflection	73
Comprehension	73
Challenges	73
Enjoyment	73
Mindset	73
Community	73
WEEK 8 OF 16: INTERMEDIATE CIRCUITS	74
Lesson Overview	74
Disciplinary Core Ideas	74
Learning Target(s)	74
Technical Skills	75
Life Skills	75
Essential Questions	75
Key Vocabulary	75
Additional Resources	75
Videos	75
Documents	75
Other	75
Depth of Knowledge Levels Addressed	76
Barriers To Learning	76
Anticipatory Set	76
Independent Practice	76
Final Assessment, Project, or Product	76
Lesson Materials (per student)	76
WEEK 8 OF 16: INTERMEDIATE CIRCUITS	77

Lesson Plan	77
Review	77
Step 1: Introducing The Breadboard (15 minutes)	77
Vocabulary	78
Step 2: Creating Our First Breadboard Circuit (15 minutes)	79
Step 3: Introducing The Barnabas Noggin and Resistance (20 minutes)	81
Vocabulary	82
Step 4: Going Further With Resistance (15 minutes or as time permits)	83
Step 5: Going Even Further With Resistance (5 minutes or as time permits)	84
WEEK 8 OF 16: INTERMEDIATE CIRCUITS	85
Reflection	85
Comprehension	85
Challenges	85
Enjoyment	85
Mindset	85
Community	85
WEEK 9 OF 16: INTRODUCTION TO CODING: THE BLINKING LIGHT CHALLENGE!	86
Lesson Overview	86
Disciplinary Core Ideas	86
Learning Target(s)	86
Technical Skills	86
Life Skills	86
Essential Questions	86
Key Vocabulary	87
Additional Resources	87
Videos	87
Documents	87
Other	87
Depth of Knowledge Levels Addressed	87
Barriers To Learning	87
Anticipatory Set	87
Independent Practice	87
Final Assessment, Project, or Product	87
Lesson Materials (per student)	87
WEEK 9 OF 16: INTRODUCTION TO CODING: THE BLINKING LIGHT CHALLENGE!	88
Lesson Plan	88
Review	88
Step 1: Connecting The Barnabas-Bot (10 minutes)	89

Vocabulary	89
Step 2: Getting Started With Ardublock (5 minutes)	90
Vocabulary	92
Step 3: Ardublock Basics (5 minutes)	92
Vocabulary	93
Step 4: Our First Program, Programming The LED (20 minutes)	94
Vocabulary	96
Step 5: Blinking The LED (20 minutes)	96
Vocabulary	97
WEEK 9 OF 16: INTRODUCTION TO CODING: THE BLINKING LIGHT CHALLENGE	99
Reflection	99
Comprehension	99
Challenges	99
Enjoyment	99
Mindset	99
Community	99
WEEK 10 OF 16: MORSE CODE WITH THE LED	100
Lesson Overview	100
Disciplinary Core Ideas	100
Learning Target(s)	100
Technical Skills	100
Life Skills	100
Essential Questions	100
Key Vocabulary	100
Additional Resources	100
Videos	100
Documents	100
Other	100
Depth of Knowledge Levels Addressed	101
Barriers To Learning	101
Anticipatory Set	101
Independent Practice	101
Final Assessment, Project, or Product	101
Lesson Materials (per student)	101
WEEK 10 OF 16: MORSE CODE WITH THE LED	102
Lesson Plan	102
Review	102
Step 1: Blinking At Different Speeds (25 minutes)	102

Step 2: What Is Morse Code (10 minutes)	104
Vocabulary	105
Step 3: Our First Morse Code Message (20 minutes)	105
Vocabulary	107
WEEK 10 OF 16: MORSE CODE WITH THE LED	109
Reflection	109
Comprehension	109
Challenges	109
Enjoyment	109
Mindset	109
Community	109
WEEK 11 OF 16: ASSEMBLING THE ROBOT	110
Lesson Overview	110
Disciplinary Core Ideas	110
Learning Target(s)	110
Technical Skills	110
Life Skills	110
Essential Questions	110
Key Vocabulary	110
Additional Resources	110
Videos	110
Documents	110
Other	110
Depth of Knowledge Levels Addressed	110
Barriers To Learning	111
Anticipatory Set	111
Independent Practice	111
Final Assessment, Project, or Product	111
Lesson Materials (per student)	111
WEEK 11 OF 16: ASSEMBLING THE ROBOT	112
Lesson Plan	112
Step 1: Attaching The Servo Motors	112
Vocabulary	113
Step 2: Frontplate Assembly	114
Step 3: Attaching The Backplate	115
Step 4: Attaching The Arms	116
Step 5: Attaching The Head	117

WEEK 11 OF 16: ASSEMBLING THE ROBOT	118
Reflection	118
Comprehension	118
Challenges	119
Enjoyment	119
Mindset	119
Community	119
WEEK 12 OF 16: THE BUZZER	119
Lesson Overview	120
Disciplinary Core Ideas	120
Learning Target(s)	120
Technical Skills	120
Life Skills	120
Essential Questions	120
Key Vocabulary	120
Additional Resources	120
Videos	120
Documents	120
Other	120
Depth of Knowledge Levels Addressed	121
Barriers To Learning	121
Anticipatory Set	121
Independent Practice	0
Final Assessment, Project, or Product	121
Lesson Materials (per student)	121
WEEK 12 OF 16: THE BUZZER	122
Lesson Plan	122
Step 1: Building The Buzzer Circuit (10 minutes)	122
Step 2: How Does The Buzzer Work (5 minutes)	123
Step 3: Making Tones With The Buzzer (45 minutes)	123
Vocabulary	126
WEEK 12 OF 16: THE BUZZER	127
Reflection	127
Comprehension	127
Challenges	129
Enjoyment	129
Mindset	129

Community	129
WEEK 13 OF 16: INTRO TO SERVOS	130
Lesson Overview	130
Disciplinary Core Ideas	130
Learning Target(s)	130
Technical Skills	130
Life Skills	130
Essential Questions	130
Key Vocabulary	130
Additional Resources	130
Videos	130
Documents	130
Other	130
Depth of Knowledge Levels Addressed	130
Barriers To Learning	130
Anticipatory Set	131
Independent Practice	131
Final Assessment, Project, or Product	131
Lesson Materials (per student)	131
WEEK 13 OF 16: INTRO TO SERVOS	132
Lesson Plan	132
Step 1: How Does The Servo Motor Work? (5 min)	132
Vocabulary	132
Step 2: Connecting The Motor (10min)	132
Step 3: Getting Our Robot To Wave (45 min)	133
Vocabulary	136
WEEK 13 OF 16: INTRO TO SERVOS	136
Reflection	136
Comprehension	136
Challenges	137
Enjoyment	137
Mindset	137
Community	137
WEEK 14 OF 16: SERVO MOTORS CONTINUED	138
Lesson Overview	138
Disciplinary Core Ideas	138
Learning Target(s)	138

Technical Skills	138
Life Skills	138
Essential Questions	138
Key Vocabulary	138
Additional Resources	138
Videos	138
Documents	138
Other	138
Depth of Knowledge Levels Addressed	138
Barriers To Learning	138
Anticipatory Set	139
Independent Practice	139
Final Assessment, Project, or Product	139
Lesson Materials (per student)	139
WEEK 14 OF 16: SERVO MOTORS CONTINUED	139
Lesson Plan	140
Step 1: Wiring The Other Two Motors (10 min)	140
Step 2: Experimenting With The Motors (20 min)	140
Step 3: Moving Multiple Motors (30 min)	141
WEEK 14 OF 16: SERVO MOTORS CONTINUED	144
Reflection	144
Comprehension	144
Challenges	145
Enjoyment	145
Mindset	145
Community	145
WEEK 15 OF 16: ADVANCED CODING	146
Lesson Overview	146
Disciplinary Core Ideas	146
Learning Target(s)	146
Technical Skills	146
Life Skills	146
Essential Questions	146
Key Vocabulary	146
Additional Resources	146
Videos	146
Documents	146
Other	146

Depth of Knowledge Levels Addressed	147
Barriers To Learning	147
Anticipatory Set	147
Independent Practice	147
Final Assessment, Project, or Product	147
Lesson Materials (per student)	147
WEEK 15 OF 16: ADVANCED CODING	148
Lesson Plan	148
Step 1: Subroutines (60 min)	148
Vocabulary	150
WEEK 15 OF 16: ADVANCED CODING	150
Reflection	150
Comprehension	150
Challenges	151
Enjoyment	151
Mindset	151
Community	151
WEEK 16 OF 16: PUTTING IT ALL TOGETHER	152
Lesson Overview	152
Disciplinary Core Ideas	152
Learning Target(s)	152
Technical Skills	152
Life Skills	152
Essential Questions	152
Key Vocabulary	152
Additional Resources	152
Videos	152
Documents	152
Other	152
Depth of Knowledge Levels Addressed	153
Barriers To Learning	153
Anticipatory Set	153
Independent Practice	153
Final Assessment, Project, or Product	153
Lesson Materials (per student)	0
WEEK 16 OF 16: PUTTING IT ALL TOGETHER	154
Lesson Plan	154

Step 1: Putting It All Together (45 min)	154
Step 2: Share (15 min)	154
WEEK 16 OF 16: PUTTING IT ALL TOGETHER	154
Reflection	154
Comprehension	154
Challenges	155
Enjoyment	155
Mindset	155
Community	155
Appendix A: References	155
Appendix B: Additional Resources	157

COURSE OVERVIEW

Course Description

Barnabas-Bot is an Introduction to Robotics class provides an opportunity for young students to learn Science, Technology, Engineering, and Math (S.T.E.M.) in a fun and engaging manner. This course, tailored for elementary school students, will introduce young learners to mechanical engineering, electrical engineering, and computer programming while designing and building a fully-functioning robot from the ground up. Additionally, students will practice writing and communication skills through daily journaling and interactive group activities throughout the course.

Prerequisites

This curriculum our entry level robotics program. Our students may be tinkerers who have built mini projects in the past but that is not absolutely necessary. We do however hope you come with an open heart and an open mind to explore the many facets of engineering, robotics, and personal growth.

Background

This curriculum was made with the nominal age of 12 years old in mind and is intended to take place over sixteen 1 hour sessions.

FULL COURSE MATERIALS LIST

Below is a list of all materials required for this course along with sources to purchase and estimated costs *per student*:

<u>ITEM</u>	<u>QTY</u>	<u>COST PER UNIT</u>	<u>SUBTOTAL COST</u>	<u>NOTES</u>	<u>HOW TO PURCHASE</u>
Barnabas-Bot Explorer Kit (Classroom Version)	1	\$42	\$42		https://squareup.com/store/barnabas-robotics
TOTAL PER STUDENT			\$42+tax		

Below is a list of all shared class materials required for this course along with sources to purchase and estimated costs.

<u>ITEM</u>	<u>QTY</u>	<u>COST PER UNIT</u>	<u>SUBTOTAL COST</u>	<u>NOTES</u>	<u>HOW TO PURCHASE</u>
Simple Robot Classroom Kit (10-Pack)	1	\$59	\$59	Enough for a class up to 30 students.	https://squareup.com/store/barnabas-robotics
Barnabas-Bot Spare Parts Kit	1	\$59	\$59	Enough for a class up to 30 students.	https://squareup.com/store/barnabas-robotics
TOTAL PER STUDENT			\$119+tax		



WEEK 1 OF 16: INTRODUCTION TO ROBOTICS

Suggested Time: 60-75 minutes

Lesson Overview

Disciplinary Core Ideas

Next Generation Science Standards

- ETS1.B: Developing Possible Solutions: Research on a problem should be carried out before beginning to design a solution. (3-5-ETS1-2)

Learning Target(s)

Technical Skills

- We will identify the basic parts of a robot. We will work together to build, learn, teach, and inspire.

Life Skills

- Teamwork
- Inspiration
- Communication

Essential Questions

1. What are the similarities and differences between robots and humans?

Key Vocabulary

1. Barnabas
2. Community
3. Encouragement
4. Inspire
5. Goals

Additional Resources

Videos

- None

Documents

- None

Other

- Scratch "[paper doll](https://scratch.mit.edu/projects/124980272/#player)" robot as demonstration:
<https://scratch.mit.edu/projects/124980272/#player>

<i>Depth of Knowledge Levels Addressed</i> <i>Level 1: Recall and Reproduction</i> <i>Level 2: Skills and Concepts</i> <i>Level 3: Strategic Thinking and Reasoning</i> <i>Level 4: Extended Thinking</i>	<i>Barriers To Learning</i> • Minimal understanding of metaphors (robot to human) • Lack of exposure to robots and their properties (e.g. They are machines; they take commands in the form of code; they are not human; they can perform repetitive action)
<i>Anticipatory Set</i> • Student definitions or examples of encouragement and community • Student understanding of functions of the human body parts: body, brain, heart, and personality/character (soul)	
<i>Independent Practice</i> • Student drawing of robots and label the 4 parts	
<i>Final Assessment, Project, or Product</i> • The labeled final drawing of the student's favorite robot.	
<i>Lesson Materials</i> ☐ Robot Model: Drawn or physical	



WEEK 1 OF 16: INTRODUCTION TO ROBOTICS

Lesson Plan

Instructor Mindset

There are a variety of ways to present new concepts to students. Think of different ways to explain new concepts and multiple methods of engaging students in learning these concepts.

Set up your classroom in a “community-building” environment (e.g. chairs arranged in a circle; carpet space to sit in a circle, etc.)

Step 1: Introducing Barnabas Robotics (10 minutes)

Project Objective

The overall goal of all Barnabas Robotics’ projects is to create a robot, while learning more about engineering - and about ourselves. Throughout the Barnabas-Bot project, students will explore the various disciplines of engineering--including mechanical, electrical, hardware, and software engineering--which each play a critical role in robotics. Additionally, students will have the opportunity to learn more about themselves and their peers, as the project also promotes teamwork, communication, and self-reflection.

Take the time at the very beginning of the project to explain these goals to your students to set the tone and expectation for the rest of the project.

Creating a Culture of Encouragement

The name “Barnabas” means “Son of Encouragement.” The Barnabas-Bot curriculum provides opportunities for students to be an encouragement to their peers and to themselves. This is especially important because there are parts of the robot-building process that may be challenging. Creating a culture of encouragement in the classroom will motivate students to try and persevere, even when they encounter frustration during the robot-building process.

Do not assume that all students understand what “encouragement” means or how they can be encouraging to themselves or others. You can initiate a classroom discussion about what “encouragement” means by prompting students to share examples of encouragement they have witnessed or experienced in their own lives.



Try making a game out of naming examples of community, perhaps a "Family Feud"-style game (e.g. "Survey Says...")

Step 2: Connecting the Disciplines (40 minutes)

Background Knowledge

Robotics is a cross-disciplinary field in which electrical engineering, mechanical engineering, and software programming concepts are applied to create a moving system (or a "robot"). Below is a brief explanation of how each of the three disciplines plays a part in robotics:

1) *Mechanical Engineering* is a branch of engineering that focuses on the design, construction, and use of machines and robots. By definition, a robot has to have a moving part -- like an arm, which has joints at the shoulder and the elbow so that the arm can move back and forth. The mechanical component of a robot may be a motor, lever, or gear that helps the robot move. A mechanical engineer may also design the exterior of the robot, which helps to protect the internal parts of the robot.

2) *Electrical Engineering* is a branch of engineering that focuses on the design, use, and distribution of electricity. A robot must have a power source from which to draw electricity. A robot's power source may be a battery or an electrical outlet. Just as the human heart pumps blood to the rest of our organs for our body to function, a battery or power source for a robot "pumps" electricity to the rest of the robot's body parts so that the robot can move and function. In order for electricity to reach the rest of the robot's body parts, a robot has a network of wires and circuits through which electricity can travel. This is similar to how human bodies have veins and arteries for blood to travel through to reach other organs.

To process information, the human body has a central nervous system consisting of a brain and a network of nerves and muscles. Likewise, a robot needs a "brain" to tell its body parts to move and function. The "brain" of the robot is the computer chip or circuit board.

3) *Computer Science* is the study of how computers function and process information. In order for humans to move their arms, the brain must send a message to the arm telling it to move. This message is similar to a computer program/code. The program/code dictates the way a robot behaves and functions. This is essentially the robot's personality.

Building a robot requires collaboration amongst a team of individuals with varying skills. Students going through the Barnabas-Bot curriculum will get a taste of each of the above engineering disciplines and work in collaboration with others (teacher, parent, mentor, classmate, etc.) as they build their robots from the ground up.

Discussion

Open up the robotics discussion by asking what the similarities and differences are between a human and a robot. Throughout this discussion, students may name certain parts of a robot, or of a human, to help them compare and contrast the two. As the students give examples of either similarities or differences, write them down on a whiteboard, putting some emphasis on the parts of either humans or robots, as this will segway nicely into the next part of the discussion.



Connect the Dots: Draw a stick-figure body. Add a heart, a brain, and soul (encircled gingerbread man or woman) to the body. Next, write the engineering disciplines (Mechanical, Electrical, Computer (Hardware), and Software) next to the body. Have the students draw a line connecting the engineering discipline, which corresponds most closely to the *function* of the listed body part.

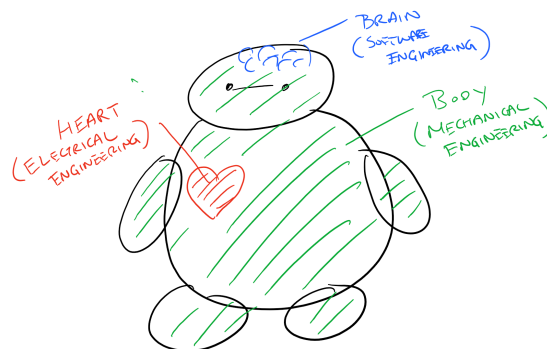
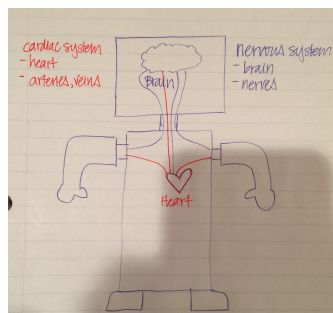


"Is This A Robot?"

Ask the students if a commonly used appliance is a robot. For example, "Is a washing machine a robot?" Have the students plead their case one way or the other for why the object is or is not a robot. The goal of this exercise is not to *correctly* identify whether the appliance is a robot or not. Rather, the goal should be to guide the discussion toward using the four parts of a robot to assess whether the object is a robot or not. Have the students attempt to match the appliance in question to that criteria of a robot (body, heart, brain, personality).



Consider drawing a robot during the parts of a robot discussion. Add the individual parts to it as they come up, like in the examples below:





REFLECTION

Comprehension

1. What are the 4 major parts of any robot?
2. What type of engineering is related to each of those parts?
3. What are we going to learn about and how are we going to learn?

Challenges

What was hard in today's class?

Enjoyment

What was fun about today's class?

Mindset

How can *you* help make the class a good learning community?

Community

How can what you learned impact those around you?



Supporting Materials

Free BrainPop video about Robots:

- <https://www.brainpop.com/technology/computerscience/robots/> (Video available in Spanish and French only with paid language-specific BrainPop log-in)

Examples of robots

- “Big Hero 6” when Baymax became evil: <https://goo.gl/UX3XEs>
- In “Toy Story” when Buzz Lightyear was set to Spanish mode: <https://youtu.be/iG9VLxlccx4>
- “The Last Bastion” by Blizzard Entertainment about a robot that was coded for evil but turns good: <https://goo.gl/jsvCV6>

Videos of conductive pens explaining open and closed circuits

- <https://www.youtube.com/watch?v=FAC3kqzWm4g> (0:25-0:50)
- <https://www.youtube.com/watch?v=ZYofuygntWA>



WEEK 2 OF 16: TEAMWORK THROUGH ROBOTICS

Suggested Time: 60-75 minutes

Lesson Overview

Disciplinary Core Ideas

Next Generation Science Standards

- 3-5-ETS1-1. Define a simple design problem reflecting a need or a want that includes specified criteria for success and constraints on materials, time, or cost.
- ETS1.A: Defining and Delimiting Engineering Problems: Possible solutions to a problem are limited by available materials and resources (constraints). The success of a designed solution is determined by considering the desired features of a solution (criteria). Different proposals for solutions can be compared on the basis of how well each one meets the specified criteria for success or how well each takes the constraints into account. (3-5-ETS1-1)
- 3-5-ETS1-3. Plan and carry out fair tests in which variables are controlled and failure points are considered to identify aspects of a model or prototype that can be improved.
- ETS1.C: Optimizing the Design Solution: Different solutions need to be tested in order to determine which of them best solves the problem, given the criteria and the constraints. (3-5-ETS1-3)
- Science is a Human Endeavor: Most scientists and engineers work in teams. (4-PS3-4)

Learning Target(s)

Technical Skills

- We will build a simple robot using homemade materials.

Life Skills

- Communicating Effectively
- Making Decisions
- Teamwork
- Problem Solving
- Confidence Building
- Preparation

Essential Questions

1. What is necessary of all community members to achieve a common goal?

Key Vocabulary

1. Servo Motor
2. Servo motor Controller

Additional Resources

Videos

- Trash-Bot solutions:
 - <https://www.youtube.com/watch?v=-K1fE6fdUlw>
 - <https://www.youtube.com/watch?v=RM-kgLlhk>
 - <https://www.youtube.com/watch?v=3EYNjyEtKho>
 - <https://www.youtube.com/watch?v=b37fPzDEjO8>

Documents

- None

Other

- None

Depth of Knowledge Levels Addressed

Level 1: Recall and Reproduction

Level 2: Skills and Concepts

Level 3: Strategic Thinking and Reasoning

Level 4: Extended Thinking

Barriers To Learning

- Lack of fine motor skills
- Minimal understanding of angles and force
- Inability to work with others in a team setting

Anticipatory Set

- Students see the function of a servo motor (it moves 180 degrees). Ideally, projected under document camera.
- Objective of the challenge: For ALL teams to knock the cup placed 8" away from the robot off the edge of the table
- Setting of norms for collaboration within the team and with other teams

Independent Practice

- None

Final Assessment, Project or Product

- Ability of the robot to knock the plastic cup placed 8" away from the robot off the edge of the table

Lesson Materials

- ❑ A ziploc bag of common household materials for each group (materials may vary from group to group). For example:
 - ❑ Pipe Cleaner x 1
 - ❑ Strip of Paper x 1
 - ❑ Paper Clip x 1
 - ❑ Toothpick x 3

- ☐ Popsicle Stick x 1
- ☐ Toilet Paper Roll x 1
- ☐ 1 Roll of Masking Tape
- ☐ 1 Servo Motor per Group
- ☐ 1 Servo Motor Controller for every 3-4 groups of 3-4 students (groups must share)
- ☐ 4 AA-batteries + battery holder for every 3-4 groups of 3-4 students
- ☐ 1 cup (paper, plastic, or styrofoam) for every 3-4 groups of 3-4 students (to be knocked over; groups must share)
- ☐ 1 ruler for every 3-4 groups (must share)
- ☐ Pencil/Pen
- ☐ Optional: document camera



WEEK 2 OF 16: TEAMWORK THROUGH ROBOTICS

Lesson Plan

Review

1. What are the four parts of a robot? What type of engineer specializes in each part of the robot?

The Trash-Bot Challenge (45 minutes)

Project Overview

- **Objective:** The goal of the Trash-Bot Challenge is to work in teams to construct a robot that will knock over a cup from at least 8 inches away.
- **Method:** Each team will be given a bag of miscellaneous household objects with which to construct their Trash-Bots. Not all teams will be given the same parts. However, each team may choose to barter and/or share with one another team to accomplish the goal if they so choose.
- **Learning Mindset:** There is a more important common goal of all groups succeeding. This is a great opportunity for the students to experience a more collaborative mindset than a competitive one.

Vocabulary

- **Servo Motor:** A motor that allows for precise control of angular position. Our servo motors only have 180° range of motion.
- **Servo Motor Controller:** The brain of this robot. A controller capable of instructing the motor to move to an angle. The controller has multiple modes which alter the behavior of the motor.

Step 1: Forming Teams (10 minutes)

Group students into teams of two or three. Introduce and explain the main parts of the project:

- **The body:** Each group will be given a bag of common household items. The students will need to get creative with these to accomplish the goal of the challenge.
- **Servo motor:** This is a common motor used in hobby robotics. It has a range of motion of 180 degrees (half a circle).
- **Servo motor controller:** This device is what controls our motor.
- **4 x AA battery:** This is what powers the motor.

This is a good time to review the parts of a robot described in the last lesson. The parts provided all resemble things previously mentioned:

- The Body: The household items + servo motor
- Heart: The batteries
- Brain: The servo motor controller
- Personality: You cannot see it, but it is the programming on the brain. It tells the motor when to spin and how much to spin.

Step 2: The Design Process (10 minutes)

Before building, it is always a good idea to plan out what we intend to build. This is called the design process. The design process plays a very important role in overall creative process. Design not only focuses on making things *functional*, but making them *beautiful* as well.

Each team should collaborate to come up with at least one design plan, which incorporates the parts they have been provided.

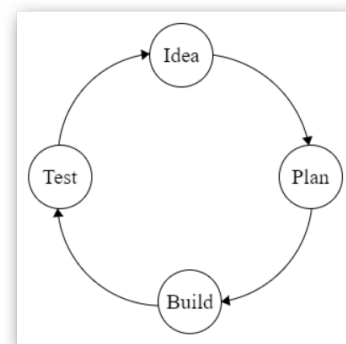
Step 3: The Build Process (40 minutes)

After a team has completed its design plan, they can start building. You can periodically have the kids bring their design up to test with the brain and battery. Encourage students to provide a hypothesis of what will happen before they activate their Trash-Bot. Ask the students to observe what actually happens.



Regardless of a group's success or failure, you should encourage them to redesign their robots. For those who succeeded, challenge them to knock the cup over from further away. For those who failed, ask critical questions: Did the robot stay together? How did you fasten parts of the robot together? Did the robot move? Which part moved? How did it move? Can that motion be used to complete the task? Can things be added to the current design to complete the task or is an overhaul of the design necessary?

The students need to understand that the design process is not linear, it is circular with the best creators iterating their projects countless times.



<http://madebyevan.com/fsm/>



REFLECTION

Comprehension

1. Why is the design process important to the overall process of making?
2. How would you describe the design process? (is it a straight line?)
3. Describe the challenges you had while making your robot.

Challenges

What some things about today that were challenging?

Enjoyment

What was fun about today's lesson?

Mindset

What did you learn about yourself? What would you like to improve?

Community

How might the knowledge you have gained impact those around you?



WEEK 3 OF 16: COMPUTER-AIDED DESIGN

Suggested Time: 60-75 minutes

Lesson Overview

Disciplinary Core Ideas

Next Generation Science Standards

- Define a simple design problem that can be solved through the development of an object, tool, process, or system and includes several criteria for success and constraints on materials, time, or cost. (3-5-ETS1-1)
- Connections to Engineering, Technology, and Applications of Science: Influence of Engineering, Technology, and Science on Society and the Natural World
Engineers improve existing technologies or develop new ones. (4-PS3-4)

Learning Target(s)

Technical Skills

- We will learn to design our robot's body using Computer-Aided Design (C.A.D.).

Life Skills

- Communicating Effectively
- Problem Solving
- Time Management
- Preparation
- Inspiration

Essential Questions

1. What challenges can Computer-Aided Design (C.A.D.) present for a beginning user?
2. Why would someone use a computer for design rather than creating a blueprint by hand?

Key Vocabulary

1. Dimensions
2. Axis
3. Plane
4. Rotation
5. Translation
6. Isometric

Additional Resources

Videos

- Show students simulation of cube rotating on its axis:
<https://www.youtube.com/watch?v=gBg4-IJ19Gg> dead link

- [Designing Barnabas-Bot in Onshape tutorial](#)
- [Putting the "B" back on the front plate](#)
- [How to put the dimension back on your front plate](#)

Documents

- None

Other

- [Onshape](#)
- [Barnabas Bot template file](#)

Depth of Knowledge Levels Addressed

Level 1: Recall and Reproduction

Level 2: Skills and Concepts

Level 3: Strategic Thinking and Reasoning

Level 4: Extended Thinking

Barriers To Learning

- Minimal understanding of 3-D concepts and geometric solids
- Minimal understanding of abstract perception
- Minimal understanding of angles (0°, 90°, 180°, 270°)
- Difficulty visualizing the possible axes of a 3D object

Anticipatory Set

- Students play with a physical geometric solid - rotating and translating the objects on an axis in varying degrees and talking about how the perception of the object changes
- Set the norm for frustration tolerance

Independent Practice

- Students work on their individual files to put the robot together and to design the front plate

Final Assessment, Project, or Product

- Completed at the end of the next lesson is a customized Barnabas Bot with a production ready STL and PDF file.

Lesson Materials

- Computer
- Computer Mouse (recommended)
- Internet connection (required)
- Onshape.com website
- NOTE: teacher must create an onshape.com account ahead of time for the entire class to use. All students will need access to the password.
- Whiteboard or poster (for posting the class onshape.com username and password)



WEEK 3 OF 16: COMPUTER-AIDED DESIGN

Lesson Plan

Review

1. Review the four parts of a robot and how it relates to the human body
2. What is the difference between design and build?

Step 1: What Is CAD? (15 minutes)

Computer-Aided Design (or CAD) is used by engineers across many disciplines. Specifically, when building a robot, CAD is used by a mechanical engineer to create the frame or body of the robot. CAD software allows us to create very precise designs with perfect lines and shape, that would be difficult to sketch by hand. CAD is also advantageous because the blueprint is stored digitally and can be accessed on most devices anywhere in the world at a moment's notice. With an electronically-stored blueprint, an engineer can then create the design using a machine and material of their preference. Our robots will be 3D-printed using plastic.



Quick 2-minute discussion of how the perspective changes and followed by a check for understanding



Hold a piece a paper behind your back that has a secret message and ask how can the students make it possible to read with getting up from their seats. (perhaps a passcode to the computers?)

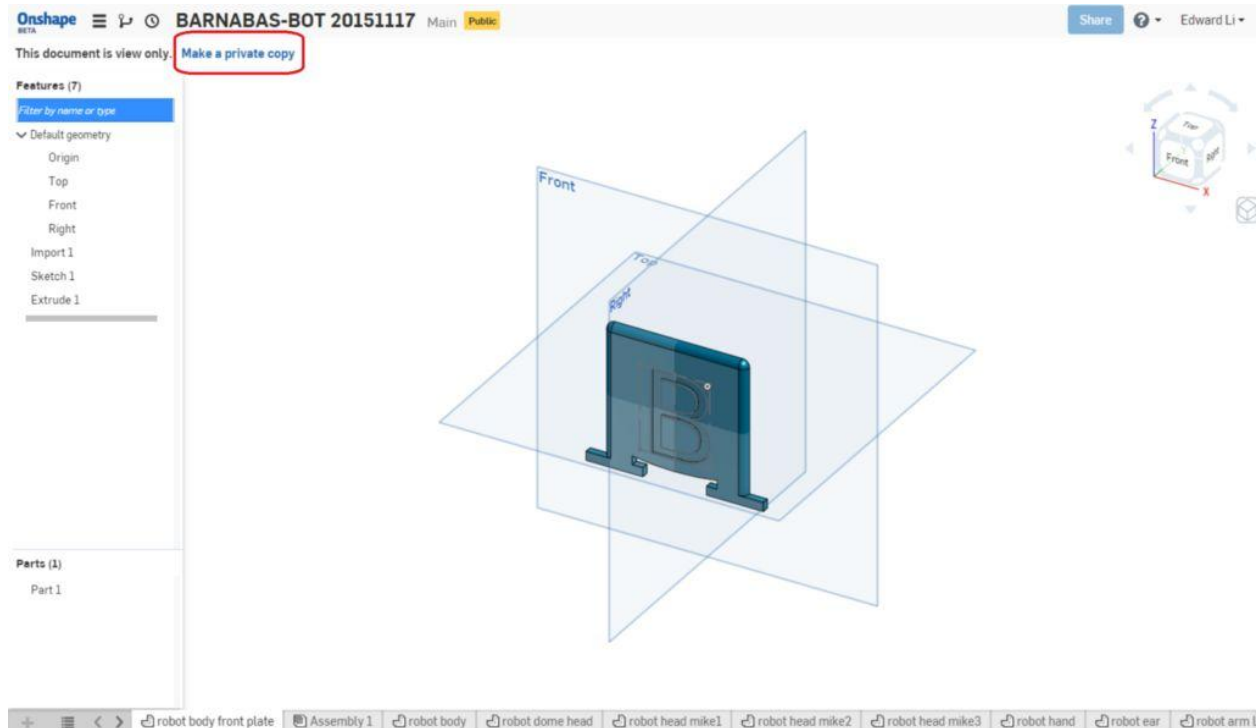
Step 2: CAD Walkthrough (45 minutes)

Preemptive Vocabulary

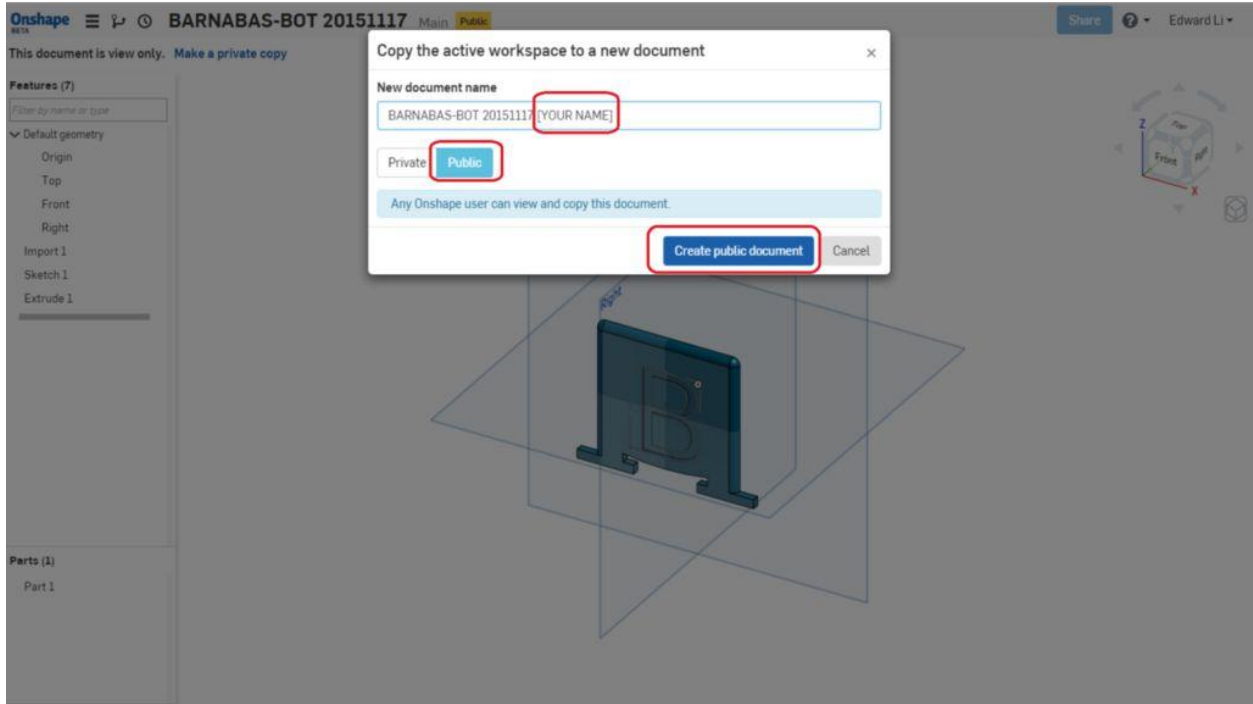
- L-CLICK: Clicking with the left button of your mouse or mouse pad.
- R-CLICK: Clicking with the right button of your mouse or mouse pad.

Onshape is a very convenient CAD tool. It uses your internet browser and saves all your files to a cloud server. Because of this we do not have to worry about installing software or misplacing files. After creating your own Onshape account follow the instructions below:

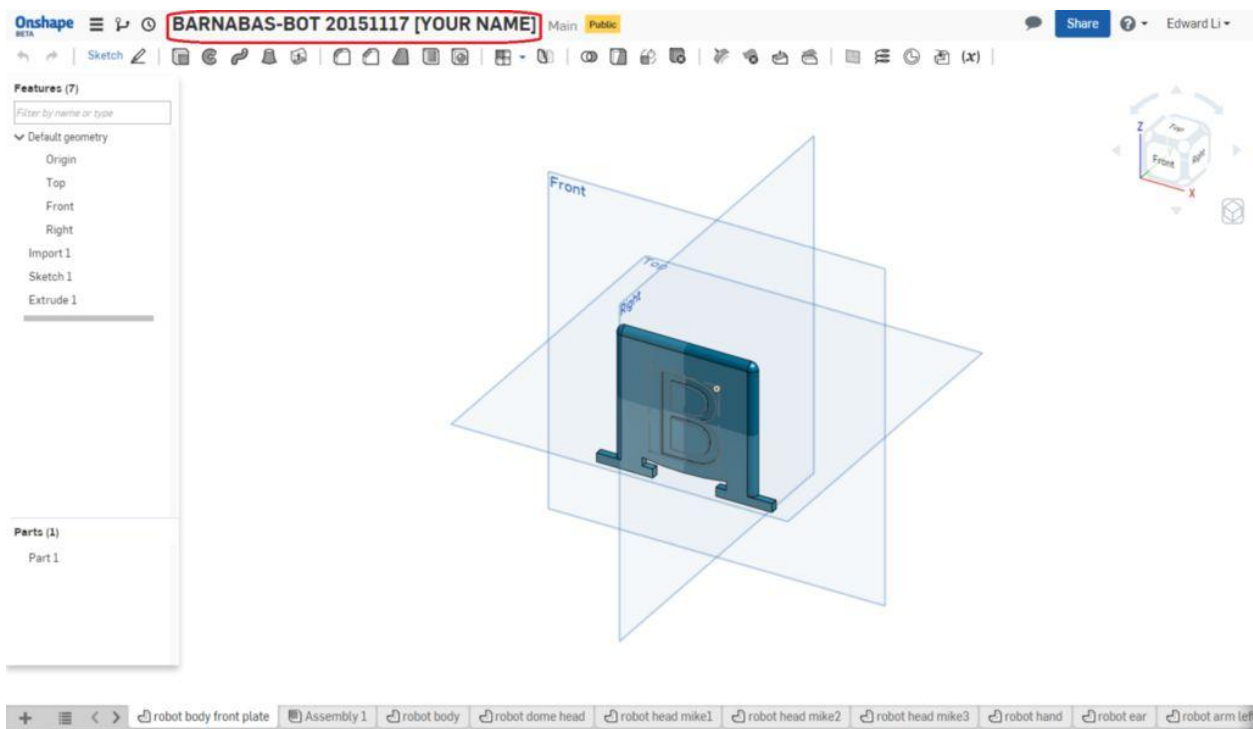
1. Once you are logged into OnShape, open the [Barnabas-Bot Template file](#).
2. Create a copy by L-CLICKing on "Make a private copy"



3. Name the project: "BARNABAS-BOT 2.3 [YOUR NAME]". Make sure to choose "Public".



4. Once you do this, a new project should open up, which will be a copy of the original project, but with this new name.

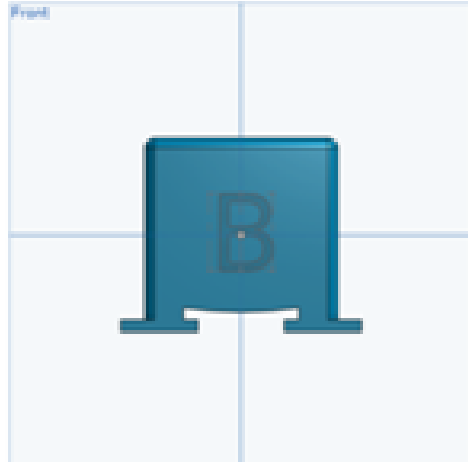


5. You are now ready to customize your robot's body!

OnShape Basics

Before diving into things you will need to become somewhat proficient with OnShape's tools. Here are a few tips on how to move stuff around in OnShape:

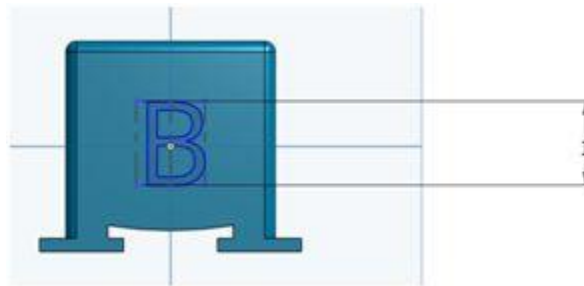
1. Open the file "robot body front plate."
 2. Rotation
 - Method 1: L-CLICK on arrows in the cube on the top left to rotate the entire perspective.
- 
- The image shows a small 3D cube in the top-left corner of the OnShape interface. The cube has three visible faces: a top face with a circular arrow, a left face with a circular arrow, and a front face with a circular arrow. The cube is used for rotating the entire perspective view.
- Method 2: R-CLICK on a part and move the mouse.
 - Method 3: L-CLICK on the "FRONT, BACK, LEFT, ETC." on the cube on the top left.
 3. Rotation
 4. Translating a part (moving it up, down and side to side)
 - Method 1: Hold down CTRL and R-CLICK on the part to move it around.
 - Method 2: L-CLICK down using the SCROLL button to move it around.
 5. Zooming
 - Use the scroll button to zoom in and out.
 - Go to "FRONT VIEW."
 - Press "F" to fit the part into your screen. This is called, "ZOOM TO FIT."



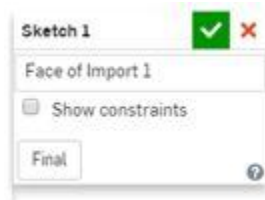
Writing on Your Robot's Body

Now we will begin the design of our own custom robot. This starts by customizing the text on the front plate (chest) of your robot. In this section we will learn both how to enlarge and how to change text:

1. Open the "robot body front plate" file in OnShape.
2. Change the size of the text
 - Double L-CLICK on the "B" until you see a number pop up on the right side of the screen



- Change the number to a larger number (like 2 or 3). The B should be larger now!
- Change the value to "0.5". The B should be smaller now!
- Click on the green check mark to save your changes



3. Move the location of the text.
 - R-CLICK on the "B" and L-CLICK on "Edit Sketch 1..."
 - L-CLICK on one of the 4 blue corners (they are kinda small, you might need to zoom in) of the rectangle around the B and drag the blue rectangle up and down



- Click on the green check mark to save your change
4. Change the text.
 - R-CLICK on the "B" and L-CLICK on "Edit Sketch 1..."
 - R-CLICK on the "B" again and L-CLICK on "Edit Text"

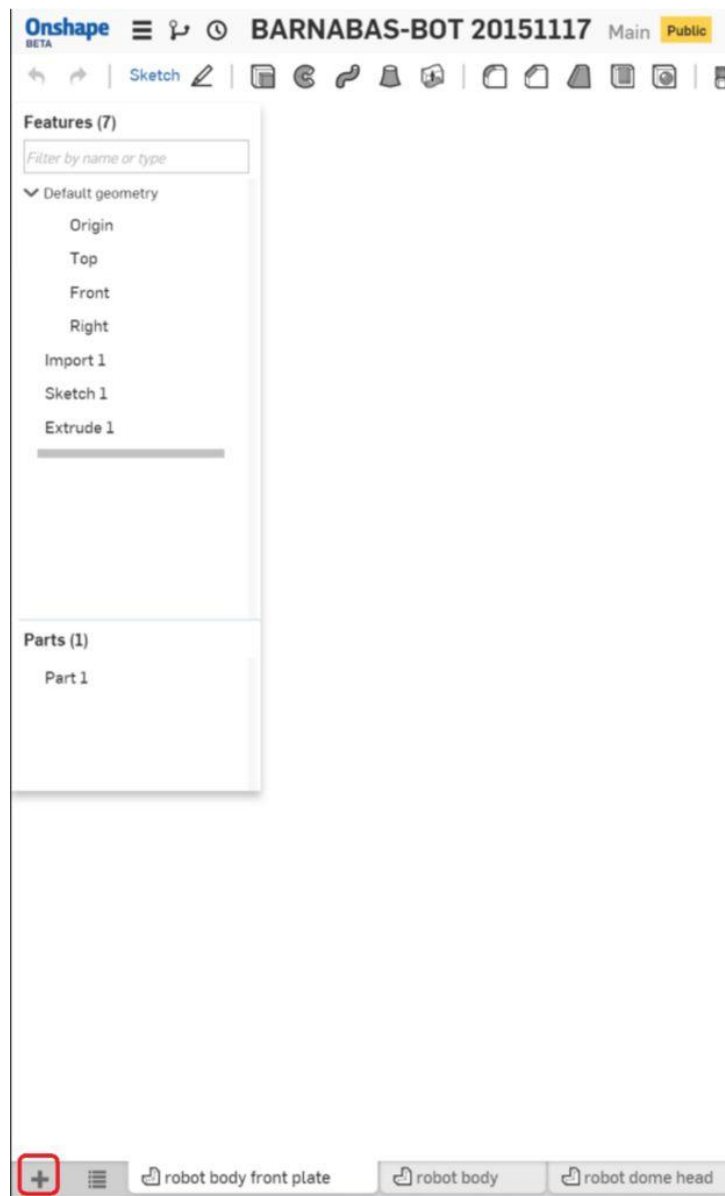


- Now change the text to "C". Click on the green check mark and see if it changes!
 - Now change the text to "BARNABAS". You'll see that it doesn't fit on the robot face. How can you make it fit? (Change the size of the text to 0.3 or less)
 - Click on the green check mark and see if it changes!
5. Now change the text to whatever you want!
 - It can be a single letter, or many letters. Play with different sizes. Make sure that all of the text fit on the robot! Remember to click on the green check mark to save your changes! Make sure to keep the size of the text larger than 0.3 so that it comes out well during printing!

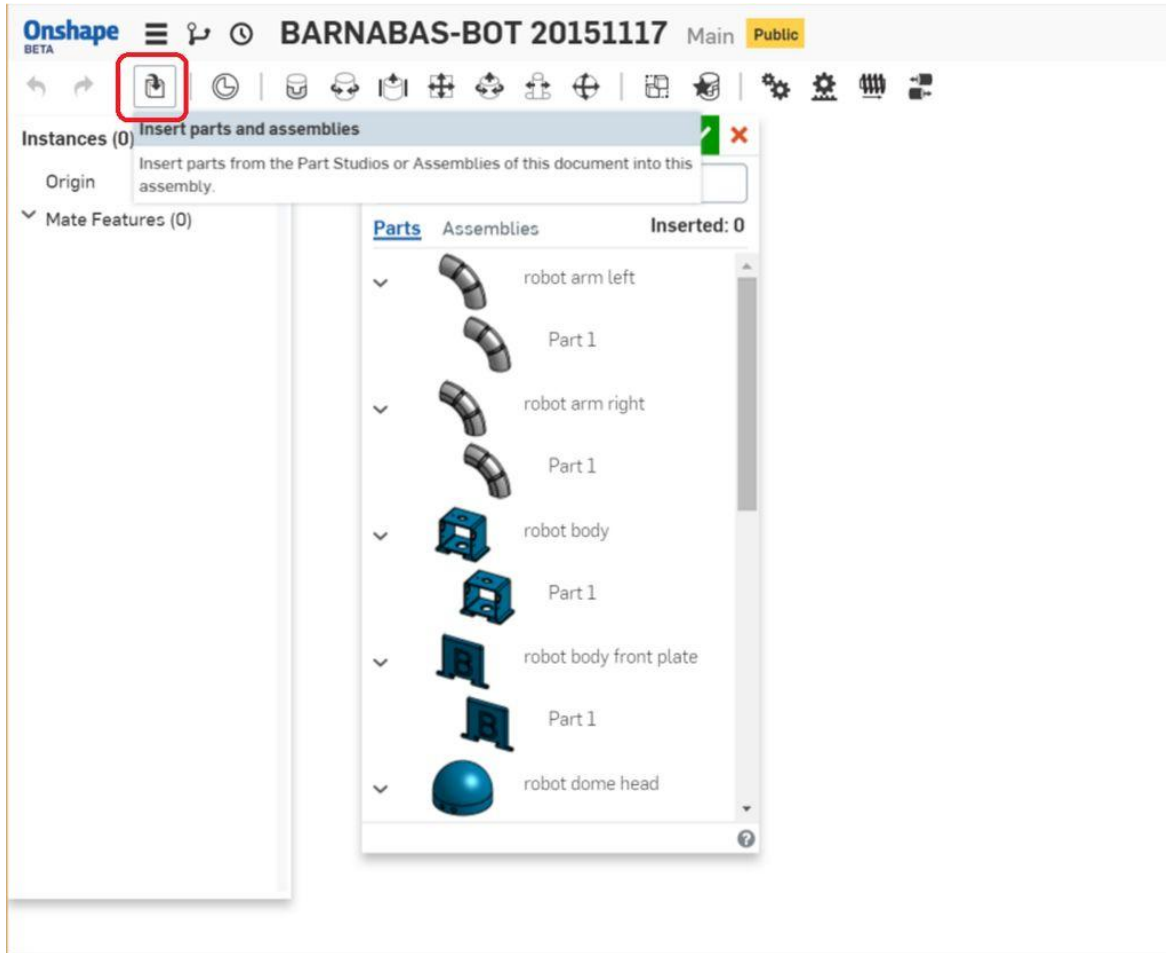
Putting Your Robot Together

This is where you will put all your robot's body parts together and create the final file that you will share with us. You will need to drag and rotate your robot's pieces into place. Keep in mind that you shouldn't be concerned with perfection. Regardless of mistakes in the assembly file, the parts will be 3D printed appropriately. Follow along with the steps below to assemble your robot:

1. Create an assembly file
 - L-CLICK on the "+" sign at the bottom left of your screen
 - L-CLICK on "Create Assembly"
 - Your assembly file should automatically open up



2. Insert your body
 - Insert your body by clicking on the insert button



- Now select the "robot body" file by selecting it in the list of files. Place it in your assembly file and click on the green check mark to save your work!
- Now go to the Front View
- 3. Insert your custom plate
 - Now insert your "robot body front plate" file.
 - Align the front plate with the body by left clicking on it and dragging it. You'll need to rotate the views to get it to align correctly



4. Insert your head
 - Using the same process as before, insert the head that you want



5. Insert your left arm and align it



6. Insert your right arm
 - You'll need to rotate it to align it correctly
 - L-CLICK on the arm to show the rotation tool
 - Click on the circle shown in the picture to rotate it



- After it is rotated, align it correctly with the hole



Vocabulary

- **Dimension:** A measurable extent, such as length, depth, or height.
- **Axis:** An imaginary line an object rotates around. Ex. the Earth has an axis that goes through its north and south pole. It rotates around this axis once every day.
- **Plane:** A flat, two dimensional object that extends infinitely far.
- **Rotation:** The movement of an object around its axis or center.
- **Translation:** The movement of an entire object in a direction. The object is not altered or rotated in any way.
- **Isometric:** Of or having equal dimensions. An isometric view, for example, will show all three dimensions equally.



If you have a projector available this is a fantastic time to make use of it. Consider going through the process of designing the robot yourself from start to finish with the students only watching. If possible they shouldn't have a computer in front of them so they are not distracted. Afterwards they can go through the design process at their own pace, and you can reiterate the finer details as needed.



It is completely reasonable to think that you will not be able to complete this lesson in one week. In a 60 minute session it is perfectly fine to only finish the front plate, possibly getting started on the assembly. Next week's content only covers about 10 to 15 minutes with that expectation. It is important that the students not feel pressured during the design process, as this is the most difficult part of the class for many.



REFLECTION

Comprehension

1. What does CAD stand for?
2. What advantages does CAD offer as opposed to designing by hand?
3. Describe, in your own words, an assembly.

Challenges

What some things about today that were challenging?

Enjoyment

What was fun about today's lesson?

Mindset

What did you learn about yourself? What would you like to improve?

Community

How might the knowledge you have gained impact those around you?



WEEK 4 OF 16: ADVANCED C.A.D.

Suggested Time: 60-75 minutes

Lesson Overview

Disciplinary Core Ideas

Next Generation Science Standards

- Define a simple design problem that can be solved through the development of an object, tool, process, or system and includes several criteria for success and constraints on materials, time, or cost. (3-5-ETS1-1)
- Connections to Engineering, Technology, and Applications of Science: Influence of Engineering, Technology, and Science on Society and the Natural World Engineers improve existing technologies or develop new ones. (4-PS3-4)

Learning Target(s)

Technical Skills

- We will learn to design our robot's body using Computer-Aided Design (C.A.D.).

Life Skills

- Setting Goals
- Preparation
- Perseverance

Essential Questions

1. What challenges can Computer-Aided Design (C.A.D.) present for a beginning user?
2. What is perseverance?
3. Why is it important to practice perseverance?
4. Why should we take the opportunity to teach others what we know?
5. Does teaching help others? ourselves?

Key Vocabulary

Additional Resources

Videos

- Add shapes <https://youtu.be/chbKu-L9G5U>

Documents

- None

Other

- None

Depth of Knowledge Levels Addressed

Level 1: Recall and Reproduction

Level 2: Skills and Concepts

Level 3: Strategic Thinking and Reasoning

Level 4: Extended Thinking

Barriers To Learning

- Minimal understanding of angles (0°, 90°, 180°, 270°)
- Difficulty visualizing the possible axes of a 3-D object

Anticipatory Set

- Students play with a physical geometric solid - rotating and translating the objects on an axis in varying degrees and talking about how the perception of the object changes
- Set the norm for frustration tolerance

Independent Practice

- Students work on their individual files to put the robot together and to design the front plate

Final Assessment, Project, or Product

- Rotating and inserting the additional parts to the body

Lesson Materials

- ☐ Computer
- ☐ Internet connection
- ☐ Onshape.com website
- ☐ NOTE: teacher must create an onshape.com account ahead of time for the entire class to use. All students will need access to the password.
- ☐ Computer mouse for each device (ideal, but optional)
- ☐ Whiteboard or poster (for posting the class onshape.com username and password)



WEEK 4 OF 16: ADVANCED C.A.D.

Lesson Plan

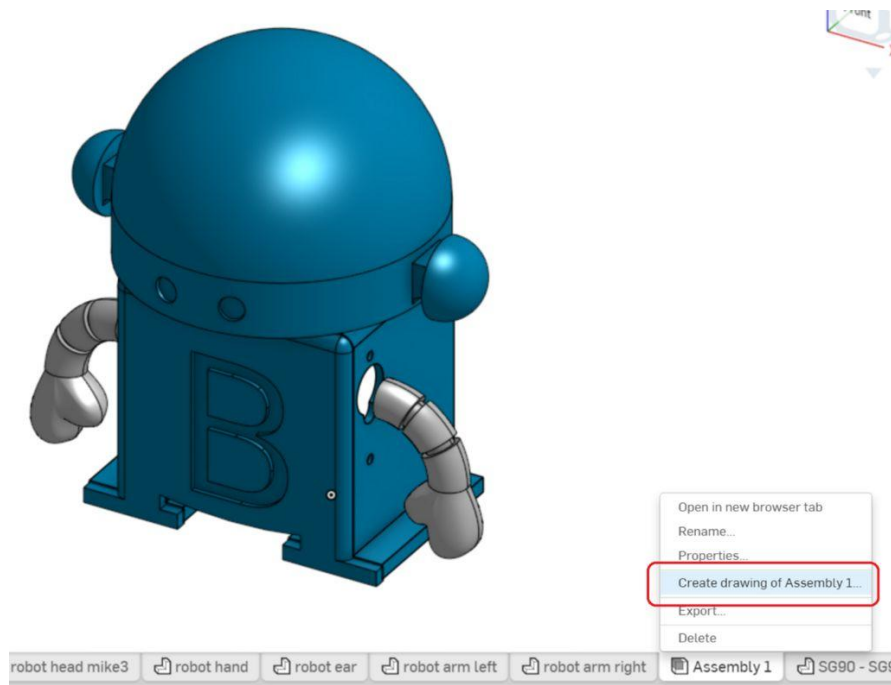
Review

1. What does C.A.D. stand for?
2. Can you explain what 3-D means?
3. What advantages does CAD give us as opposed to designing by hand?

Step 1: Creating An Assembly File (10 -15 minutes)

With the robot assembled we have the opportunity to generate a drawing of the robot. The drawing consists of four 2 dimensional views of the robot, each from a different angle. The overall drawing resembles something closer to a traditional blueprint and will be included with your robot's 3D printed parts when those are shipped to you.

1. R-CLICK on the assembly file tab that you just created
2. L-CLICK on "Create drawing of ... "



- A screen will pop up. Make sure that "Four views" is selected under "OPTIONS", and L-CLICK on "OK"

Create drawing Assembly 1 Drawing 1 ✕

Existing templates Custom template

Recently opened

- Onshape
- My templates
 - Created by me
 - Shared with me
 - Public

All ANSI ISO

Template	Document
ANSI_A_MM.dwt	Onshape ANSI Drawing Templates

[Show Onshape drawing templates](#)

Options



Four views

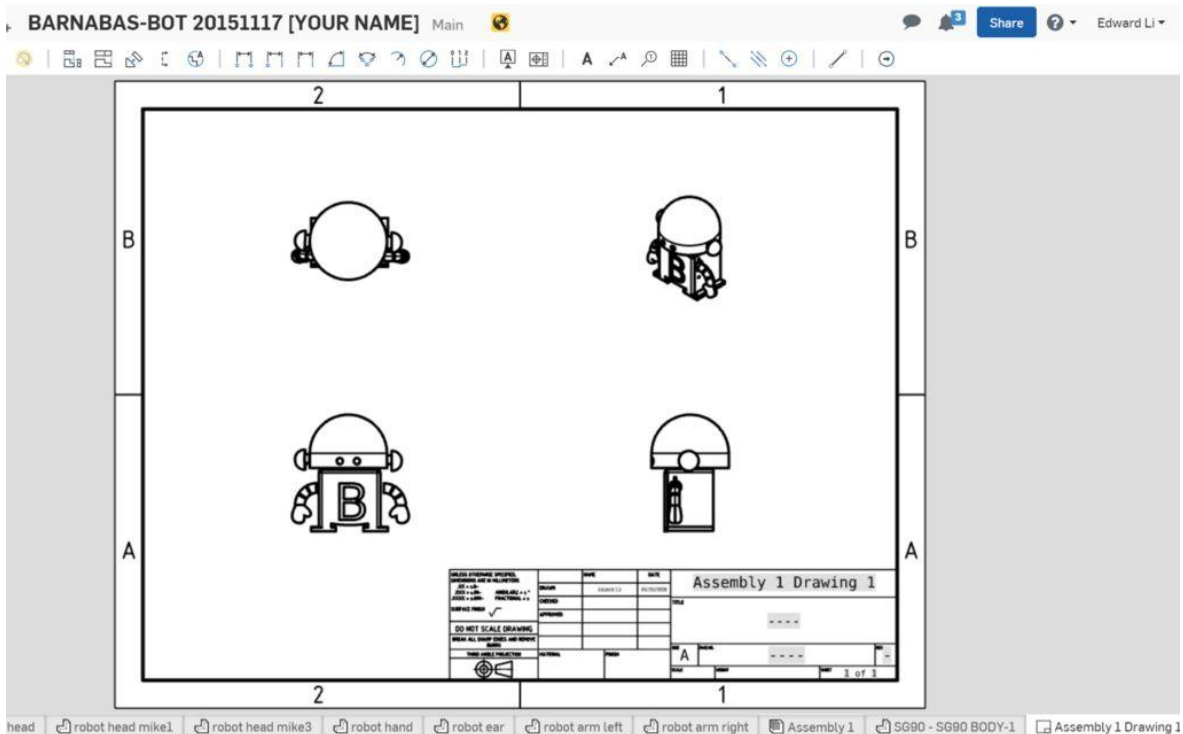


No views

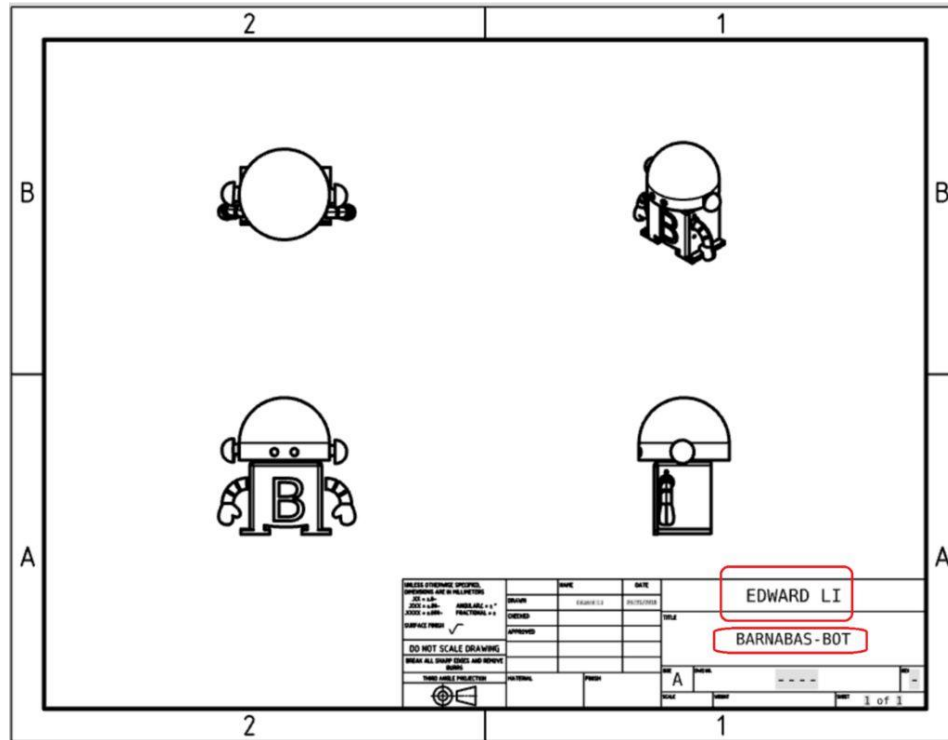
OK

Cancel

- A drawing will be automatically created.



- Double L-CLICK on the text boxes to edit the text. Write your name as well as your robot's name.



- You are done!

Share your robot with us!

This step is necessary if you would like us to 3-D print your robot for you.

- [Directions on how to share file on OnShape](#)
- Share it with the following email address: "info@barnabasrobotics.com". Once you share it with us, we will provide a confirmation within 24 hrs. Now just sit back and relax as we start putting your robot kit together!



Create a Custom Part (BONUS)

<https://youtu.be/f5TutKGCvLM> Dimensions and Constraints

<https://youtu.be/2kt-yNn5HU>



REFLECTION

Comprehension

1. What constraints or challenges did you have to keep in mind in designing your robot?
2. Can you describe the four views in the drawing using last week's vocabulary?

Challenges

What some things about today that were challenging?

Enjoyment

What was fun about today's lesson?

Mindset

What did you learn about yourself? What would you like to improve?

Community

Why should we take the opportunity to teach others what we know? Does it help others?
Does it help ourselves?



WEEK 5 OF 16: 3-D PRINTING

Suggested Time: 60-75 minutes

Lesson Overview

Disciplinary Core Ideas

Next Generation Science Standards

- Constructing Explanations and Designing Solutions: Apply scientific ideas to solve design problems. (4-PS3-4)
- Science is a Human Endeavor: People's needs and wants change over time, as do their demands for new and improved technologies. (3-5-ETS1-1)
- Science affects everyday life. (4-PS3-4)

Learning Target(s)

Technical Skills

- We will learn the parts and properties of a 3-D printer.
- We will learn about industrial design and the importance of form and function in design.

Life Skills

- How to have a discussion

Essential Questions

1. What are the limitations of 3-D printing?
2. What is the importance of form and function in regards to the design process?
3. How do you suspect the robot was printed? (How were the pieces oriented on the printer?)

Key Vocabulary

1. Filament
2. Extruder
3. Bed
4. Cartridge

Additional Resources

Videos

- Video: [3-D Printer and How it is Made](#)
- Video: [3-D Printer in Action](#)

Documents

- None

Other

- None

Depth of Knowledge Levels Addressed

Level 1: Recall and Reproduction

Level 2: Skills and Concepts

Level 3: Strategic Thinking and Reasoning

Level 4: Extended Thinking

Barriers To Learning

- There shouldn't be any barriers in this lesson.

Anticipatory Set

- Hand students a few 3-D printed objects, if you have them. What do they notice about the objects? What process do they think is needed to print the objects?

Independent Practice

- Asking/answering questions regarding the 3-D printer and the process of printing 3-D objects.

Final Assessment, Project, or Product

- Drawing of robot with color schemes to be used for painting in the next session

Lesson Materials

- ☐ Portable 3-D printer
 - ☐ If a 3-D printer is not available for students to see, opt for the video
- ☐ Engineering journal
- ☐ Writing utensil



WEEK 5 OF 16: 3-D PRINTING

Lesson Plan

Review

1. What CAD operations did we learn to do last week?

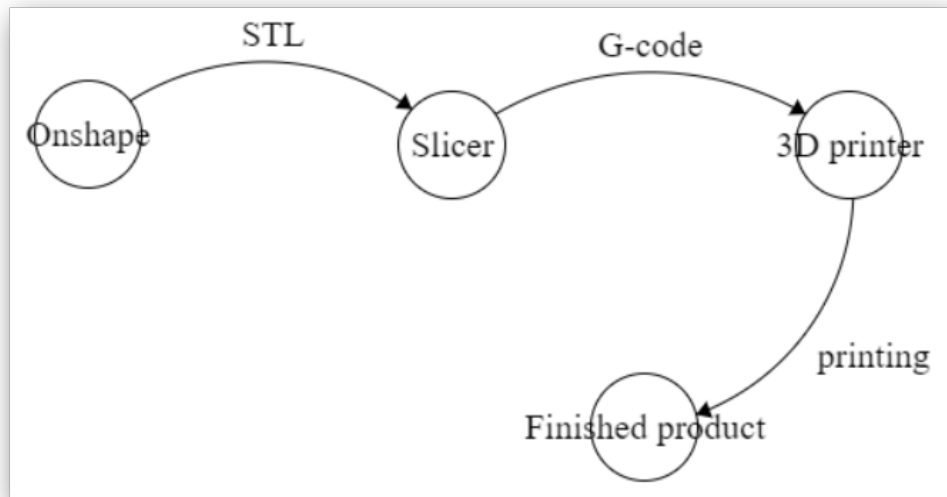
Step 1: Introduction To 3-D Printing (25 minutes)

Over the last two lessons the process of 3-D printing has been mentioned, but I think it is time to explicitly ask: “What is 3-D printing?”. That question is much more complicated than it originally seems. We can begin to answer it by drawing similarities and differences between a 3-D printer and your paper printer at home. The big difference is that a 3-D printer will print layers one on top of another, accumulating height while doing so. Using an analogy we can say that the 3-D printing process is much like writing your name on a cake, the icing sits atop the rest of the cake.

3-D printers are used by engineers in the field to quickly build designs. They are used as rapid prototyping machines more often than not, giving engineers the opportunity to design, build and test an idea in the same day. This means that while 3-D printers are not always creating the finished product, they are still crucial to the design process.

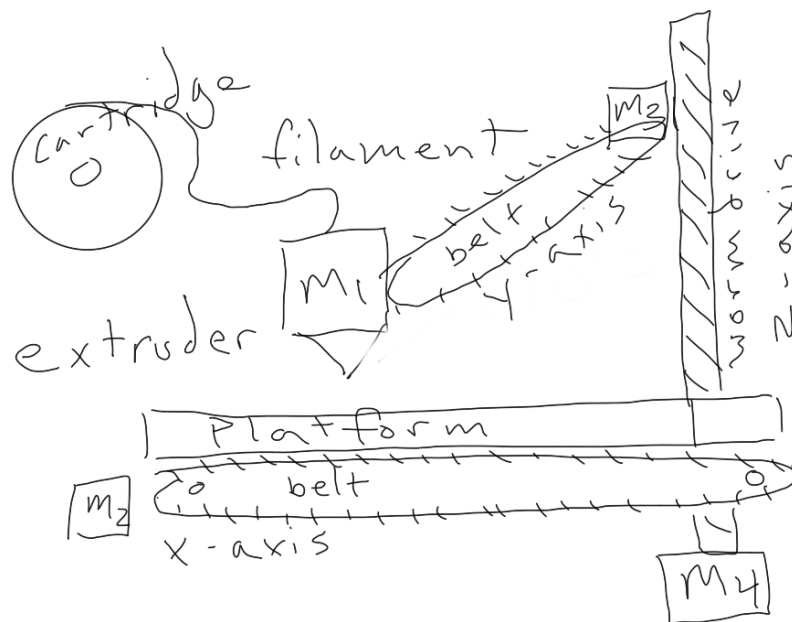
The process of getting a 3-D file to the printer isn’t as simple as you may suspect. It is not as easy as throwing the file we designed into a 3-D printer and turning it on. The CAD file we designed previously can be exported from Onshape as what is called an STL file. STLs are a common format for 3-D files. Unfortunately for us, a 3-D printer does not accept STL files. Instead they are made to accept g-code files. A g-code file is a file made up of many two dimensional drawings which will ultimately be printed one on top of the other to create the finished product. A g-code file tells the 3-D printer the specifics of how it should move and how much plastic to extrude at any one time.

What is needed is some way of converting an STL file to a g-code file. This can be done with a *licer* program. A slicer will take the 3-D STL file and slice it into each 2-D piece. The overall process is shown in simplified form below:



<http://madebyevan.com/fsm/>

Draw a 3-D printer on the whiteboard, making sure to label important components:



Going over the significant parts:

Vocabulary

- Filament: A long string of plastic that is fed into the extruder where it is melted and used to print an object.

- Extruder: The component responsible for heating and printing the plastic material of the filament.
- Cartridge: Where the filament is held.
- Platform: Where the object is printed.

Have the students attempt to point out these parts on the 3-D printer. Perhaps also ask them questions such as: How many motors do you see? What does each motor do?

Step 2: Industrial Design (20 minutes)

Industrial design is a term that refers to two different aspects of a manufactured product. The first is the object's usefulness (function). The second is the object's beauty (form). When engineers design products for the public they need to carefully consider the role of the product and determine how much of their effort goes into the form of the product and how much goes into the function of the product. Some things do not need to look pretty, they just need to work. Others need to draw people's attention to have a chance in the market.

Lead the students in a discussion about common items (especially electronics) and ask the students if they think that form or function is more important for that item.



Have the class discuss reasons they like either Samsung or Apple phones. Group each reason into two categories: function or beauty. Explain the story of the [Samsung and Apple dispute over rounded edges](#). This activity enforces that looks do matter!



Have your students design a smartphone for themselves and one for someone they know, like a relative, friend or sibling. How do those two phones differ? Perform the same exercise with something simpler, like a sandwich. How is the sandwich you made for yourself different from the sandwich you made for the other person? Why are they different?

Hand each student the drawing of their robot as well as a box with the robot parts inside. The box should contain everything except the custom 3-D printed parts. You can hand those out next time. The drawing of their robot can be placed inside the box under the top.



REFLECTION

Comprehension

1. Explain the significance of 3-D printing in the design process.
2. What are some of the 3-D printers components and what does each of them do?
3. Describe the process that begins with designing an object in a CAD program and ends with the 3-D printed object.
4. Why is the form or beauty of a product important to engineers?

Challenges

What some things about today that were challenging?

Enjoyment

What was fun about today's lesson?

Mindset

What did you learn about yourself? What would you like to improve?

Community

Why should we take the opportunity to teach others what we know? Does it help others?
Does it help ourselves?



WEEK 6 OF 16: TIME TO DECORATE

Suggested Time: 60-75 minutes

Lesson Overview

Disciplinary Core Ideas

Learning Target(s)

Technical Skills

- Industrial Design

Life Skills

- Creativity

Essential Questions

Key Vocabulary

Additional Resources

Depth of Knowledge Levels Addressed

Level 1: Recall and Reproduction
 Level 2: Skills and Concepts
 Level 3: Strategic Thinking and Reasoning
 Level 4: Extended Thinking

Barriers To Learning

Anticipatory Set

- Students will spend lesson painting or coloring their robots.

Independent Practice

- Paint/color robot based on color scheme the student designed last week.

Final Assessment, Project, or Product

- Fully decorated robot.

Lesson Materials

- ☐ Painting/coloring materials
 - ☐ Colored sharpies are a good option
 - ☐ Acrylic paint is a good option
 - ☐ Be sure to bring butcher paper or some other type of table cover.
- ☐ Engineering journal
- ☐ Writing utensil



WEEK 6 OF 16: TIME TO DECORATE

Lesson Plan

Review

1. What is industrial design?

Examining The Parts (10 min)

Hand out the custom parts, the heads and front plates, to each of the students. Have them examine the 3-D printed parts to make sure that they match with their original CAD drawing. This is a good time to also think about how each 3-D printed part was laying on the bed as it was printed. HINT: look at how the lines on the 3-D printed bots are oriented.

Hand out whatever supplies you are going to have the students use, paints or otherwise.

Creating A Story (10 min)

Explain to the students that the next class they will be able to paint/color their robot. Have the students individually come up with a design plan or color scheme with their robot.

Spend some time with the class discussing possible designs for their robots. Give the class a few sentence starters.

1. My robot's name is:
2. My robot's job is to:
3. My robot's colors are:

Each student can share their robot's design with their neighbor or class before they start decorating.

Decorating The Robot (50 minutes)

Be sure to cover the tables of the classroom in some way, we recommend butcher paper. This is an opportunity for the students to make each of their robots unique.



This is a good time to infuse some of your artistic knowledge into the class. A quick discussion of [additive](#) vs [subtractive](#) colors our [color moods](#) would be appropriate.



REFLECTION

Challenges

What was hard?

Enjoyment

What was fun?

Mindset

What did you learn about yourself? What do you need to work on?

Community

How can what you learned impact those around you?



WEEK 7 OF 16: INTRODUCTION TO CIRCUITS

Suggested Time: 60-75 minutes

Lesson Overview

Disciplinary Core Ideas

Next Generation Science Standards

- Planning and Carrying Out Investigations: Make observations to produce data to serve as the basis for evidence for an explanation of a phenomenon or test a design solution. (4-PS3-2)
- Constructing Explanations and Designing Solutions: Apply scientific ideas to solve design problems. (4-PS3-4)
- PS3.B: Conservation of Energy and Energy Transfer: Energy can also be transferred from place to place by electric currents, which can then be used locally to produce motion, sound, heat, or light. The currents may have been produced to begin with by transforming the energy of motion into electrical energy. (4-PS3-2),(4-PS3-4)
- Energy and Matter: Energy can be transferred in various ways and between objects. (4-PS3-1),(4-PS3-2),(4-PS3-3),(4-PS3-4)

Learning Target(s)

Technical Skills

- Understanding of basic electronics concepts including open-circuits, closed-circuits, short-circuits, voltage and current
- Ability to create a simple circuit with an LED using AA batteries.
- Ability to extrapolate necessary voltage for two LED circuits and beyond.

Life Skills

- Teamwork
- Freedom to fail

Essential Questions

1. What is a circuit?
2. What is the difference between open and closed circuits?
3. What is current?
4. How does voltage effect a circuit?

Key Vocabulary

1. Circuit
2. Open-Circuit
3. Closed-Circuit
4. Current
5. Voltage
6. Short-Circuit

Additional Resources

Videos

- Creating the first LED circuit:
<https://www.youtube.com/watch?v=5P-3rSK79Uo&t=15s>

Documents

- None

Other

- None

Depth of Knowledge Levels Addressed

Level 1: Recall and Reproduction

Level 2: Skills and Concepts

Level 3: Strategic Thinking and Reasoning

Level 4: Extended Thinking

Barriers To Learning

- Minimal experience with household electronics.
- No working knowledge of electricity.

Anticipatory Set

- Learn basic electrical engineering concepts and use them to create a simple LED circuit

Independent Practice

- Assembly of LED circuit with a partner. Time permitting, the assembly of a two LED circuit with three partners.

Final Assessment, Project, or Product

- Drawing the two LED circuit schematic.

Lesson Materials (per student)

- ☐ AA Battery (x1)
- ☐ AA Battery connector (x1)
- ☐ 3V LED (x1)
- ☐ Engineering Journal (x1)
- ☐ Pencil (x1)



WEEK 7 OF 16: INTRODUCTION TO CIRCUITS

Lesson Plan

Review

1. What are the four main parts of a robot?
2. Which kind of engineer works on each part of the robot?

Step 1: Electronic Basics (20 minutes)

Draw a robot. I like to draw Eve from Wall-E. Eve's battery is analogous to a human heart. Draw a heart on your robot. We cannot live without our hearts pumping blood to the rest of our bodies, and robots cannot turn on without a battery. Human hearts pump blood, similar to how robots batteries supply electrical power.

Eve's wires are analogous to a human's arteries, veins, and capillaries. Draw a full circulatory system on your robot. We cannot be healthy if our arteries are blocked or veins are weak and cannot supply appropriate blood flow to our body. Similarly, issues with wiring can seriously affect the robot even if the battery is fully charged.

Eve's current flow is analogous to human circulation. Just as blood flows in one direction in our bodies: from our heart out to arteries into capillaries in our bodies where oxygen is used to veins that carry the blood back to our heart where the oxygen can be replaced (actually the blood also flows separately to the lungs to be reoxygenated, it is written here thusly for simplicity), electricity flows through a circuit in a particular direction so that the robot works best. Just as the circulatory system makes a loop, a circuit makes a loop too.

Vocabulary

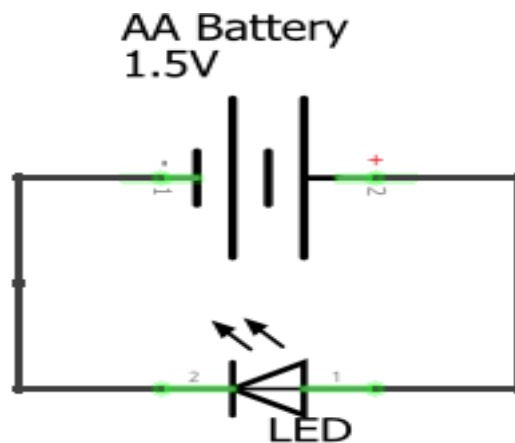
- **Closed-circuit:** A closed-circuit is when there is a complete loop in your electrical system. In this case electricity can flow.
- **Open-circuit:** An open-circuit is when the loop is broken. In this case electricity cannot flow. The robot cannot live if we have an open circuit, much like how we cannot live if blood is not circulating through our bodies.
- **Short-circuit:** A short circuit is when you have a loop, but electricity is flowing too fast. Just like blood being pumped through our bodies too quickly, too much electricity flowing through a circuit can harm the robot.



Flip the light switch in the room off and ask the students if this is an open or closed circuit. Hopefully this helps cement the idea that open-circuit = no electricity.

Step 2: Building Our First LED Circuit (1.5V) (10 minutes)

Draw the LED circuit schematic shown below. Encourage the students to determine the voltage of the circuit by examining their parts. HINT: it is written on the battery.



Explain that the voltage determines how fast electricity flows through our circuit. The higher the voltage, the faster electricity moves. Explain that electricity flowing through a circuit is called 'current,' and that there are positive and negative sides on both the battery as well as the LED. They need to line up the positives and negatives for the circuit to work.

The schematic is the plan for our circuit. You can compare it to the blueprint of a building. At this point, distribute the robotics kits to your students. Have them take out:

- AA Battery
- AA Battery holder
- LED

Have the students build the circuit with the components they just took out. Let them figure out how to correctly put the battery into the battery holder, as well as which side is positive (red) and which is negative (black). The positive end of the LED has a longer wire.



...IT DOESN'T WORK!

Your students will notice that the LED does not light up. Ask them why (HINT: the 3V LED needs at least 3 volts to turn on). At this point, use the metaphor of a waterfall to describe voltage. Explain that the taller a waterfall is, the more power you will get out of it. It is the same with voltage. More voltage, more power. Ask the students how we can get the necessary 3V to light up the LED (DO NOT use the 9V battery. Instead, combine two AA batteries!).

Vocabulary

- **Current:** Current is simply the flow of electricity.
- **Voltage:** Voltage is what creates current, or the flow of electricity through a circuit. A larger voltage will cause a faster flow of electricity through a circuit.



Not all electronic components have positive and negative ends, as we will see later. However the LED being used must be placed in the following way due to some fairly simple physics. Have you ever made a lemon or potato battery? The

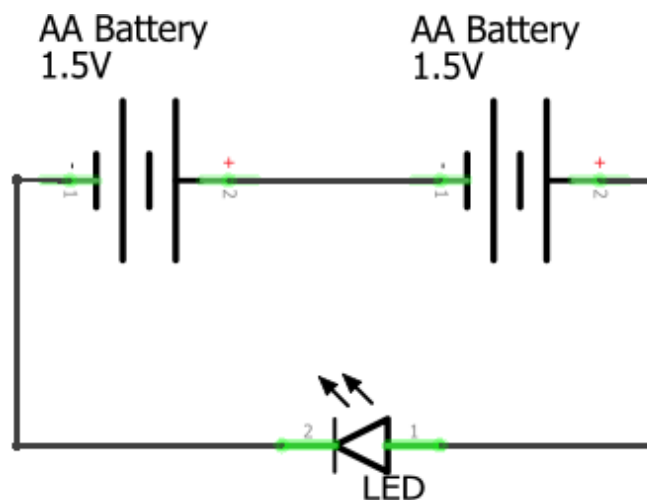
key to getting a potato battery to work is that either side of the potato must be poked with some sort of metal AND a different metal must be used for each side. When the two metals are placed side by side one seems slightly negatively charged and the other seems slightly positively charged. This is what allows the potato to generate voltage and make current travel in a specific direction. The LED likewise has two different metals at either side of it, so it prefers current to flow in one direction through it.



Before handing out the robotics kits consider giving a general discussion about safety, both for the students and for the robot. Make sure to emphasize that the electronic parts are easy to break (usually by short circuiting them) and that they need to act responsibly with them. If problems arise, I recommend confiscating the 9V batteries until they are needed. Those pose the biggest threat to student and robot safety.

Step 3: Building The 3 Volt Circuit (15 minutes)

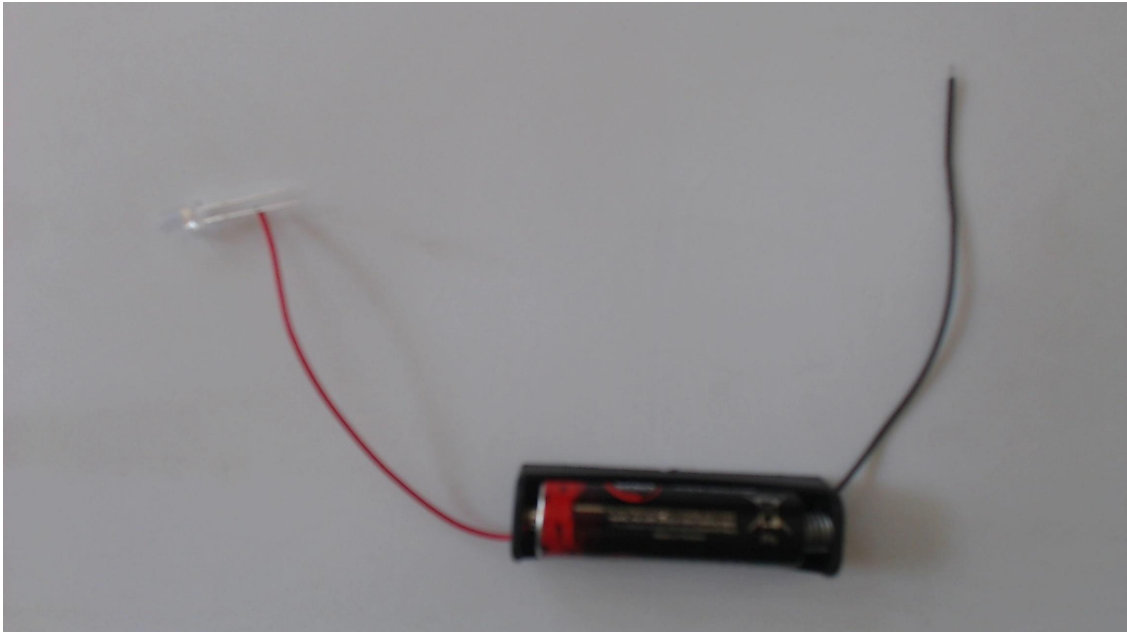
Start by drawing the 3V LED schematic.



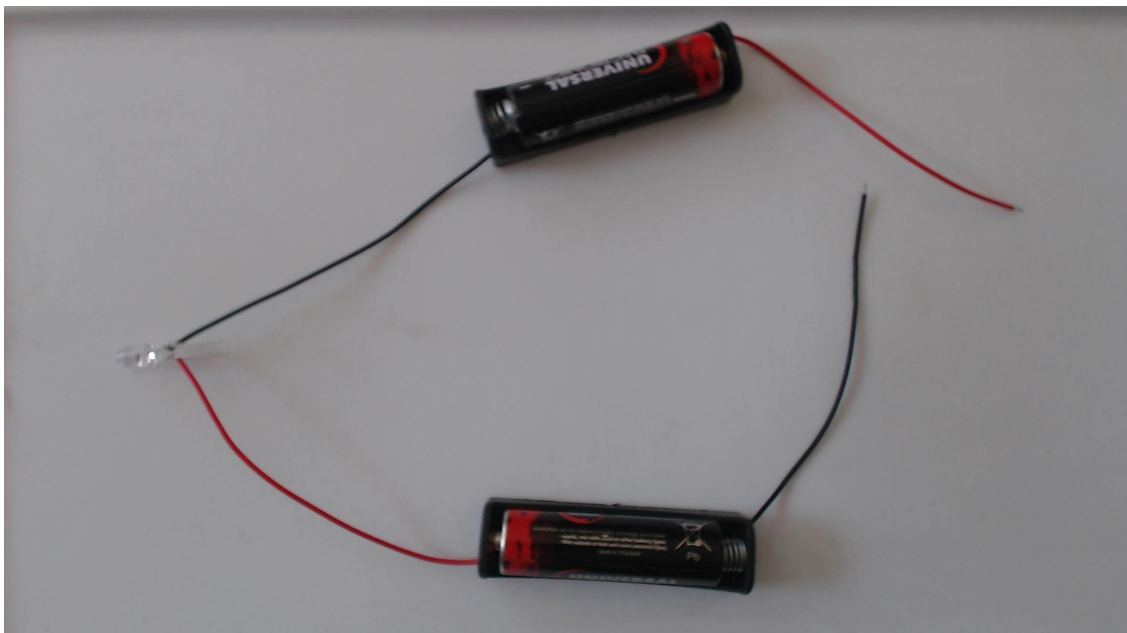
Ask the students to identify the difference between this schematic and our previous one (there are two batteries). How many total volts is the LED getting now.

Have the students work together to build the circuit. Everyone will need to find a partner to have enough batteries for the circuit. If you have an odd number of students, you as the instructor can fill in as a partner for one of your students. After giving the students an opportunity to build the circuit themselves, bring the class together and walk through building the circuit one step at a time.

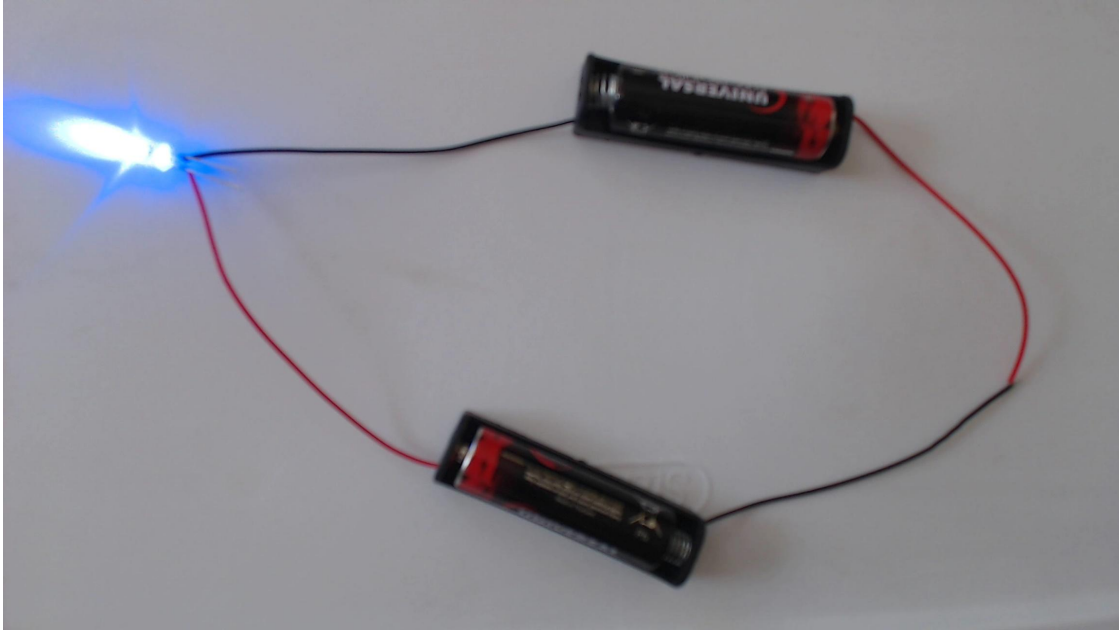
1. Connect the + wire(red) of the first battery to the + leg(the longer leg) of the LED.



2. Connect the - leg(the shorter leg) of the LED to the - wire(black) of the second battery.



3. Connect the + wire(red) of the second battery to the - wire(black) of the first battery.

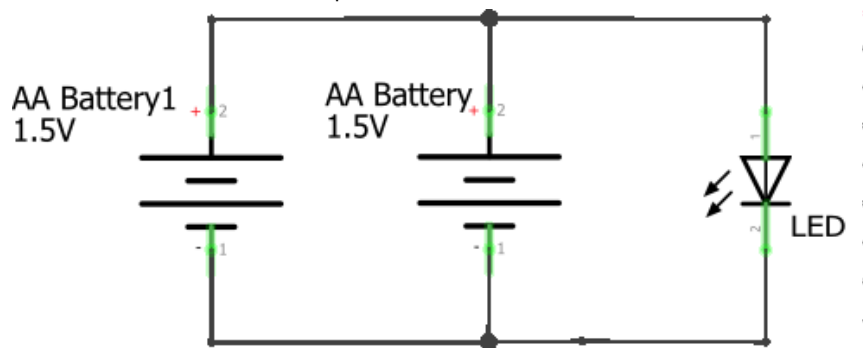


4. All finished!



Students often connect the positive terminals and negative terminals of the batteries together, rather than connecting the negative to positive (as shown in the schematic). This incorrectly creates a parallel circuit. If they do this, they will see that the LED still doesn't light up. You can take this opportunity to talk about "series" circuits versus "parallel" circuits, if you like. A "series" circuit stacks the batteries on top of each other, so that the voltage is added together (like putting two same height waterfalls on top of each other). A "parallel" circuit puts them next to each other so that the voltage stays the same, but there is more current available (like putting two same height waterfalls next to each other).

See below for a schematic for the incorrect parallel circuit.

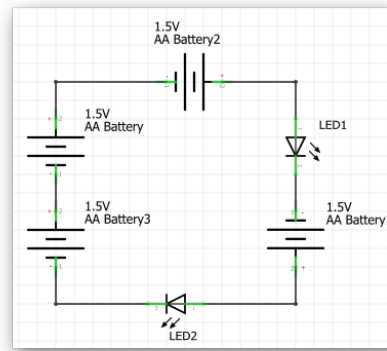
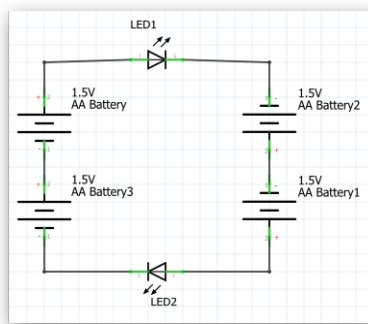
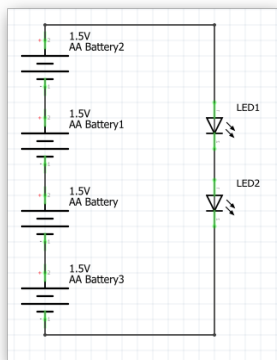


Another error is for students to attempt to use more than two batteries to light a single LED. While this will likely give a blighter glow to the LED it is likely to damage it as well.

Another common error is to connect the batteries incorrectly. Remember that The red wire of one battery must be connected to the black wire of the other battery. In a circuit where the two red wires are connected or the two black wires are connected electricity will not flow. Going back to the waterfall analogy, we can increase the height of our waterfall by connecting batteries together from negative to positive. We are taking additional steps in the same direction, up. However connecting two like ends, whether negative or positive, has the opposite effect. One battery is attempting to increase the height of the waterfall while the other is decreasing it.

Step 4: The Two LED Circuit (15 minutes or as time permits)

Now that the students have successfully made the 3V LED circuit it's time to give them a bigger challenge. Instead of pairs, have the student's get into groups of four. Between the four of them they should have enough batteries to light two LEDs. Give them the opportunity to attempt a two LED circuit on their own. Unlike the one LED circuit, this two LED circuit has multiple working schematics. See below.



You should challenge your students to build the circuit without showing them the schematics above. These are more for the benefit of the teacher, so you may be more equipped to help your students with troubleshooting issues.



After each group has successfully created the circuit, challenge them to create the schematic of the circuit they have just built. This activity is best when the schematics above have not yet been shown to them



REFLECTION

Comprehension

1. What is a circuit? What is the difference between an open-circuit and a closed-circuit.
2. Why did our initial circuit (one AA battery) not light the LED?
3. If I wanted to light 5 LEDs, how many AA batteries would I need?

Challenges

What was hard?

Enjoyment

What was fun?

Mindset

What did you learn about yourself? What do you need to work on?

Community

How can what you learned impact those around you?



WEEK 8 OF 16: INTERMEDIATE CIRCUITS

Suggested Time: 60-75 minutes

Lesson Overview

Disciplinary Core Ideas

Next Generation Science Standards

- Planning and Carrying Out Investigations: Make observations to produce data to serve as the basis for evidence for an explanation of a phenomenon or test a design solution. (4-PS3-2)
- Constructing Explanations and Designing Solutions: Apply scientific ideas to solve design problems. (4-PS3-4)
- PS3.B: Conservation of Energy and Energy Transfer: Energy can also be transferred from place to place by electric currents, which can then be used locally to produce motion, sound, heat, or light. The currents may have been produced to begin with by transforming the energy of motion into electrical energy. (4-PS3-2),(4-PS3-4)
- Energy and Matter: Energy can be transferred in various ways and between objects. (4-PS3-1),(4-PS3-2),(4-PS3-3),(4-PS3-4)
- Science is a Human Endeavor: Science affects everyday life. (4-PS3-4)

Learning Target(s)

Technical Skills

- Further comprehension of basic electronics concepts from last lesson.
- Ability to utilize breadboard when building circuits.
- Understanding of the concept of resistance and the resistor.
- Ability to identify specific pins on Barnabas Noggin used to power circuits, and understanding the Barnabas Noggin's use as a power source.

Life Skills

- Teamwork

Essential Questions

1. How does the breadboard help us make circuits?
2. How does one create a closed circuit on the breadboard?
3. What is the purpose of the resistor in

Key Vocabulary

1. Breadboard
2. Resistor/Resistance
3. Barnabas Noggin
4. Component

- our circuit?
4. Which of the two types of resistor is stronger? Is the LED dimmer or brighter with the stronger resistor.

Additional Resources

Videos

- Creating the first LED circuit:
<https://www.youtube.com/watch?v=5P-3rSK79Uo&t=15s> (second half of video uses breadboard)
- Creating the 5V LED circuit using the Barnabas noggin:
<https://www.youtube.com/watch?v=iY3Ygg6Ucq0>
- Explaining how a breadboard works:
<https://www.youtube.com/watch?v=QFm8Gkofgs8>

Documents

- None

Other

- Sparkfun breadboarding tutorial:
<https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard>
- Resistor color code calculator:
<https://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-resistor-color-code-4-band>

Depth of Knowledge Levels Addressed

Level 1: Recall and Reproduction

Level 2: Skills and Concepts

Level 3: Strategic Thinking and Reasoning

Level 4: Extended Thinking

Barriers To Learning

- Minimal experience with household electronics.
- No working knowledge of electricity.
- Poor understanding of lesson 7.

Anticipatory Set

- Learn intermediate electrical engineering concepts.
- Introduce new components and use them to create circuits.

Independent Practice

- Assembly of LED circuit using a breadboard with a partner. Time permitting, the assembly of a two LED circuit using a breadboard with three partners.
- Assembly of 5V LED circuit using breadboard, resistor, and Barnabas Noggin.

Final Assessment, Project, or Product

- Comparison of 5V circuit with each type of resistor. Based on the behavior of the circuit which resistor is stronger? Defend your answer.
- Using Ohm's Law, have each student calculate the current in their circuit with the resistor of their choice.

Lesson Materials (per student)

- ☐ AA Battery (x1)
- ☐ AA Battery connector (x1)
- ☐ 3V LED (x1)
- ☐ Jumper wires
- ☐ Resistors (470 Ohm - yellow violet brown gold and 4.7K Ohm - yellow violet red gold)
- ☐ Mini breadboard (x1)
- ☐ Barnabas Noggin (x1)
- ☐ Engineering Journal (x1)
- ☐ Pencil (x1)



WEEK 8 OF 16: INTERMEDIATE CIRCUITS

Lesson Plan

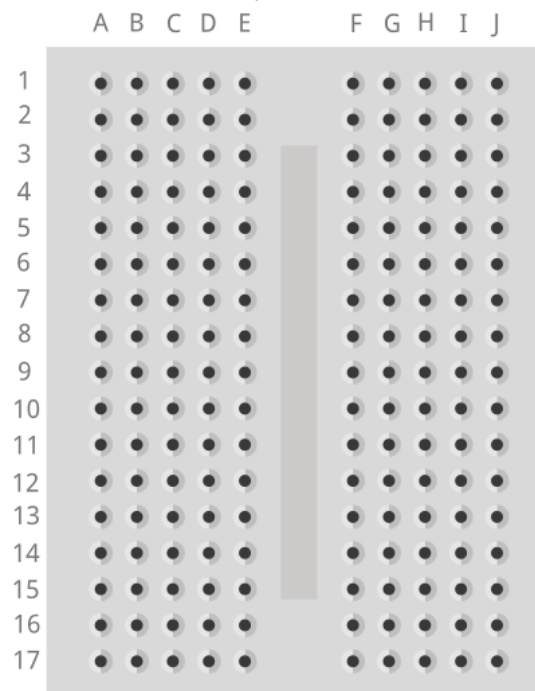
Review

1. How do open and closed circuits differ?
2. How does increasing voltage affect a circuit?

Step 1: Introducing The Breadboard (15 minutes)

In this lesson we are introduced to a very valuable tool, The breadboard. What is a breadboard?

A breadboard helps us connect our wires (analogous to blood vessels) together. We can connect our wires to the holes in the breadboard to make secure connections without needing to tie the wires together. This is useful, especially when we need to change the connections often. This is common while engineers figure out how to best make their ideas happen. Draw a picture of the miniature breadboard on the whiteboard. NOTE: You can use the 1-17 (rows) and A-J (columns) references to tell kids where to place wires as the lesson progresses.



- How many rows are there? (17)
- How many columns are there? (2 sets of 5)

In a breadboard All the holes on each row of 5 are connected to each other. By connected I mean that any two things placed into the same row will be connected to each other. The breadboard helps us make closed circuits. Test the kids to see if they understand which holes connect to one another.

Vocabulary

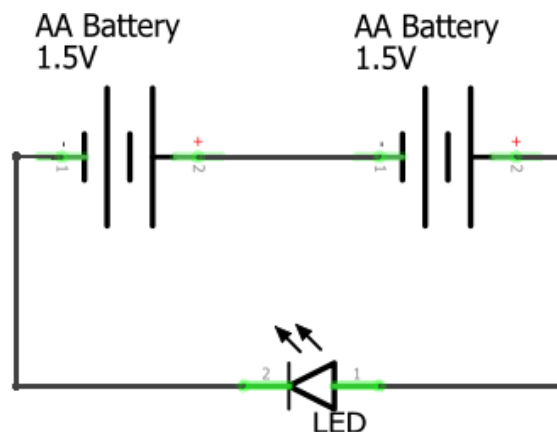
- Breadboard: A tool used to create circuits. The breadboard is able to connect two or more components together as if their were touching so long as they are on the same row.
- Component: A single element of a circuit. Ex: battery, LED.



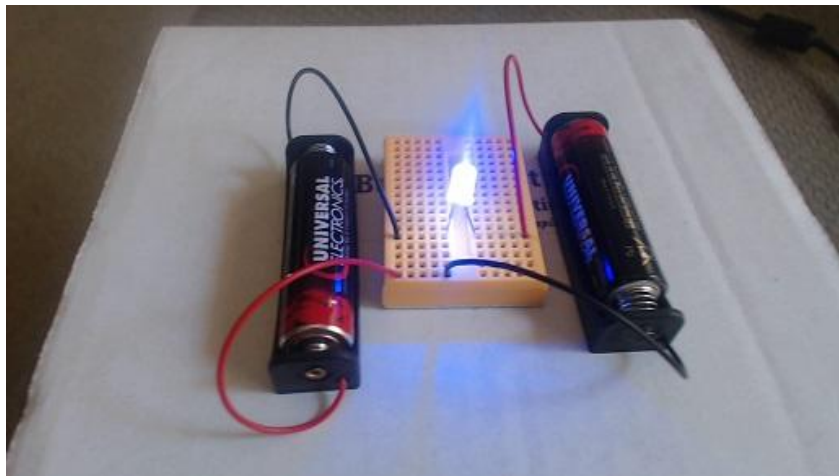
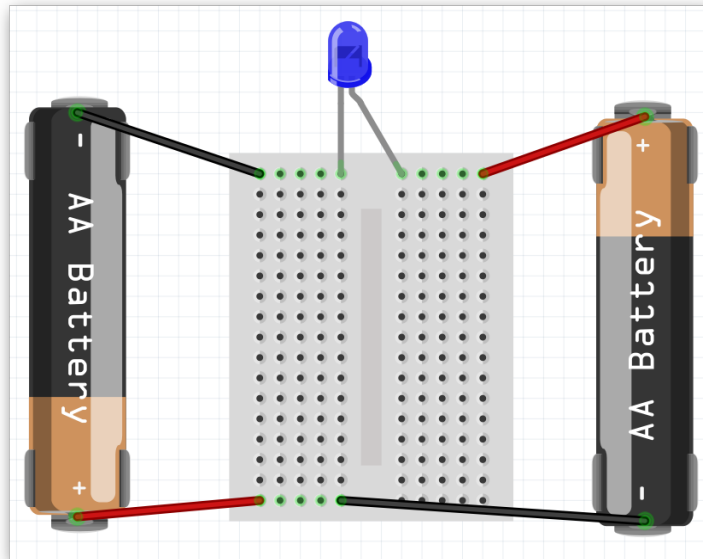
It is important that the students have a good grasp of how the breadboard works. To ensure comprehension you can draw boxes around each set of holes that are connected on your whiteboard drawing. Another method would be to remove the adhesive from the back of a breadboard (I would suggest you use a spare for this rather than using a student's) exposing the strips of metal underneath. This may help them understand why the breadboard works as it does.

Step 2: Creating Our First Breadboard Circuit (15 minutes)

Now that we understand how to use the breadboard we can recreate the 3V LED circuit from the last lesson. This time as a completely hands free circuit. Let's start by recalling the schematic for that circuit.



To create the circuit using the breadboard you need to make the same exact connections as before. A helpful tip is that any two wires/legs that were held together in the previous circuit must now be in a row together. Not necessarily in the same hole though. Also, both ends of a component should never be in the same row together. That is the same as touching the two sides together which, in the case of the battery, creates a short circuit. They should end up with a circuit that looks similar to the one below.



**BEST
PRACTICE**

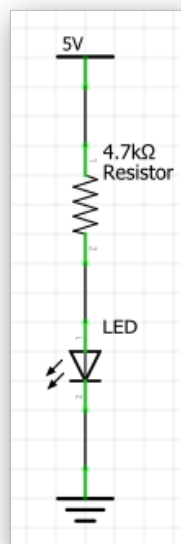
Before asking the students to create the circuit, draw the schematic on the board. Then draw a breadboard. Enlist the aid of the students in drawing batteries and the LEDs, as well as placing them in the correct spots to create a complete circuit. Once finished, follow the path of the circuit with the class and ensure that the circuit is closed.



A common mistake is to put far too many connections on the same row of the breadboard. This is generally the most common reason a group's circuit will not work. This is something you should keep an eye out for and simply remind students that only things should be in any row of the breadboard at once. Reminding them of their method of creating the circuit from last week's lesson may also help, as this is essentially the same circuit.

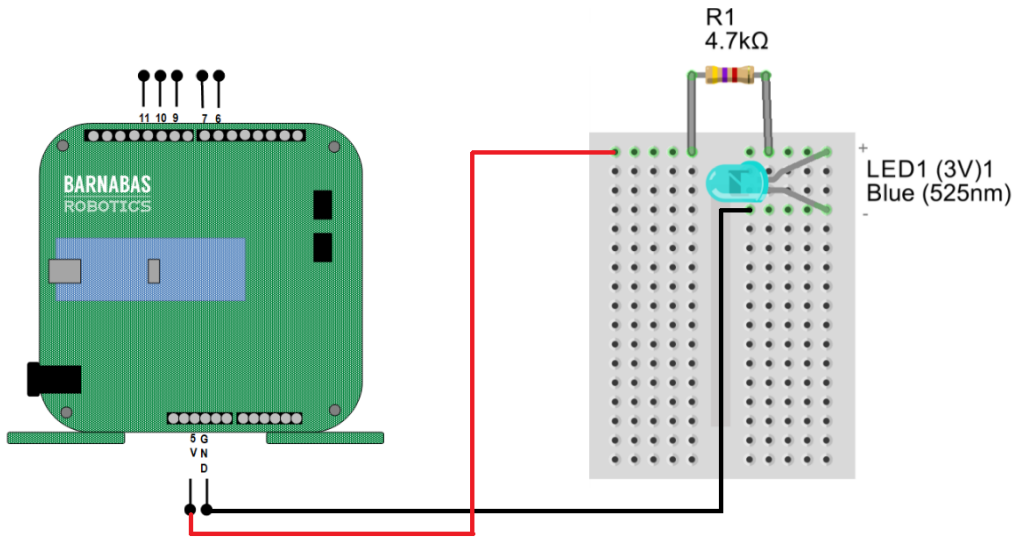
Step 3: Introducing The Barnabas Noggin and Resistance (20 minutes)

For the first time we are now going to use the Barnabas Noggin to create a circuit. With the Noggin we will be powering the circuit with 5V instead of 3V. What do you think will happen when we increase the voltage? (The LED will be brighter, and we no longer need the AA batteries) In fact, the 5V is too much. Ask what will happen if you push 5V through a 3V LED (it will burn out). Talk about what happens in our body if our heart (battery) pumps blood too fast through our veins (it may burst!). Explain that the same thing happens in electronics, so to combat that, we need to slow the blood (current) down using a resistor. Have the kids take out the 4.7K resistor (red band). Explain that a resistor is the component, and that resistance is the value (4.7K or 4,700 in this case). Draw the following schematic.



Before asking the students to attempt building this circuit on their own guide them through placing wires in the Noggin. Have them place one end of a wire (preferably red) into the 5V pin located on the Noggin. Have them place one end of another wire (preferably black) into one of the GND pins located on the Noggin (there is more than one). The other ends of both wires are

currently dangling free. These ends should be treated the same as the + and - ends of a battery. Now they should be ready to create the circuit below.



Plug the 9V battery into the jack on the Noggin and your LED should light up!

Vocabulary

- **Barnabas Noggin:** The brain of our soon to be robot! The Barnabas Noggin houses the brain of the Barnabas bot. It also contains a large amount of internal circuitry, making it somewhat related to the robot's heart as well. We will often treat the Noggin as a power source, as it can give us 5V, a voltage we are comfortable with.
- **Resistance:** The difficulty of passing an electric current through an object. If an object is very difficult to pass current through, it has a high resistance.



Sometimes your students' LEDs may burn out because of incorrect connections. This is all part of the learning process. We recommend having a working Arduino + breadboard setup that you can use to test to see if students' LEDs are in fact burned out or if there is perhaps another wiring issue.



Resistance, along with voltage and current are the three fundamental quantities in a circuit. In fact, there is an equation that relates all three of these inside of a circuit. The equation is known as Ohm's Law $V=I \times R$, where V is the voltage, R is the resistance, and I is the current. This simple equation gives us a very good understanding of the behavior of any circuit.



Have the students switch between the two different kind of resistors and observe the behavior of the LED as the resistor is changed. They should see the LED dim when the red banded resistor is in the circuit. This is the larger resistor. Because it is larger, it slows down current more, leading to the dimmer light.



It is always a good idea to have power disconnected from our circuit as we are constructing it, or making changes to it. In this case, even in the event of a short circuit, we will not damage any of our parts. It is best to build the circuit, check it for errors, then apply power.



Why is it important to color code wires? Color coding wires helps us both construct and troubleshoot our circuits. If I know that a certain color of wire always means a certain thing, it will be obvious when that wire is out of place

Step 4: Going Further With Resistance (15 minutes or as time permits)

You may have noticed the two different types of resistor that we have are differentiated based on the colors of their bands. As it turns out we can tell the exact strength of a resistor based on the color of those bands.

Color	Numeric Value	Multiplier	Tolerance	Temperature coefficient
BLACK	0	1 Ω		250
BROWN	1	10 Ω	$\pm 1\%$	100
RED	2	100 Ω	$\pm 2\%$	50
ORANGE	3	1K Ω		15
YELLOW	4	10 Ω		25
GREEN	5	100 Ω	$\pm 0.5\%$	20
BLUE	6	1M Ω	$\pm 0.25\%$	10
VIOLET	7		$\pm 0.1\%$	5
GREY	8			1
WHITE	9			
GOLD			$\pm 5\%$	
SILVER			$\pm 10\%$	

Let's walk through this chart together. In the Numeric Value column we see that each color is given a number value associated with it. These numbers are only applied to the first two colored bands on a resistor. Those two numbers are combined into one 2 digit number. For

example our 4,700 Ohm resistor has the colors yellow and violet as it's first two color bands, in that order. From this we know that resistor's value must start with the number 47, which we know it does. The third band on the resistor is the multiplier. We take the number denoted by the color of that band (in this case 100-red) and multiply that number and the 2 digit number from earlier together (47×100), giving us 4,700 Ohms.

The fourth band on a resistor is what's called it's tolerance. Tolerance is the error that one can expect in the resistor. For example, a gold tolerance band on our 4,700 Ohm resistor tells us that it may not be exactly 4,700 Ohms, but we can expect it to be within 5% of that. So the resistor could be as small as 4,465 Ohms or as big as 4,935 Ohms. The last column in the table above is labelled Temperature Coefficient. We don't need to worry about what that means, as it is only used in resistors with five bands or more.



Have the students come up with the color code for a resistor that has a strength of your choosing. If you would like, you can put the students in groups and give each group a different resistor value to work on.

Step 5: Going Even Further With Resistance (5 minutes or as time permits)

With knowledge of Ohm's law, we can calculate the current going through the circuit when either resistor is used. Ohm's law ($V=IR$) can be rewritten as $I=V/R$. Both the voltage and the resistance are known to us so we can calculate current in both cases.

- $I=5/470 \rightarrow I=0.01$ Amps (Amperes are the unit used to measure current)
- $I=5/4700 \rightarrow I=0.001$ Amps



Not all resistors are 'Ohmic' resistors, as in ones that follow Ohm's laws. An Ohmic resistor always has the same resistance, where as a non Ohmic resistor has a fluctuating resistance based on some parameter. For example if you were to measure the resistance of a light bulb while it is off, you would measure a fairly low resistance. But if you turned the lightbulb on, the resistance would skyrocket due to how much hotter the lightbulb became. If non Ohmic resistors are used in simple circuits such as this one, some interesting effects can be created.



Do not dismantle the circuits at the end of this lesson. We will be building on these circuits for the rest of class and it would be convenient

if we did not have to start from scratch each week. Just make sure that the students unplug their batteries before putting the circuits away.

REFLECTION



Comprehension

1. What is the breadboard used for? How does the breadboard work?
2. What is resistance? How does resistance affect our circuit?
3. What part(s) of the robot does the Barnabas Noggin represent?

Challenges

What was hard?

Enjoyment

What was fun?

Mindset

What did you learn about yourself? What do you need to work on?

Community

How can what you learned impact those around you?



WEEK 9 OF 16: INTRODUCTION TO CODING: THE BLINKING LIGHT CHALLENGE!

Suggested Time: 60-75 minutes

Lesson Overview

Disciplinary Core Ideas

Next Generation Science Standards

- ETS1.B: Developing Possible Solutions: Tests are often designed to identify failure points or difficulties, which suggest the elements of the design that need to be improved. (3-5-ETS1-3)
- ETS1.C: Optimizing the Design Solution: Different solutions need to be tested in order to determine which of them best solves the problem, given the criteria and the constraints. (3-5-ETS1-3)
- Energy and Matter: Energy can be transferred in various ways and between objects. (4-PS3-1),(4-PS3-2),(4-PS3-3),(4-PS3-4)

Learning Target(s)

Technical Skills

- Ability to operate a computer.
- Understanding of physical computing fundamentals.
- Understanding of cooperation between hardware and software in physical computing.
- Working knowledge of Arduino/Ardublock and how the Barnabas-Bot interfaces with the computer.

Life Skills

- Procedure

Essential Questions

1. What are the main drawbacks of Arduino?
2. Can we upload code to our bots immediately after connecting them to the computer? Why or why not?
3. Why do we have to change the pin # block from 1 to 7 on the set digital pin block.
4. Why can't we see the LED blink

Key Vocabulary

1. Arduino
2. Ardublock
3. Loop Do
4. Programmable Pin
5. Set Digital Pin
6. Delay

- without delay blocks?
5. Why is it important we use the millisecond delay instead of the microsecond delay?

Additional Resources

Videos

- Ardublock + driver install tutorial: <https://www.youtube.com/watch?v=bBSZ8REdUYA&t=1s>
- Ardublock batch install tutorial (Windows only): <https://www.youtube.com/watch?v=i2l7I9PdE2o>
- Blinking light challenge tutorial: <https://www.youtube.com/watch?v=UfxJx9LQ9u0>

Documents

- None

Other

- [Software setup on Windows](#)
- [Connecting Barnabas-Bot to a Windows PC](#)
- [Software setup on Mac](#)
- [Connecting Barnabas-Bot to a Mac](#)

Depth of Knowledge Levels Addressed

Level 1: Recall and Reproduction

Level 2: Skills and Concepts

Level 3: Strategic Thinking and Reasoning

Level 4: Extended Thinking

Barriers To Learning

- Minimal experience with computers.

Anticipatory Set

- Learn how to connect Barnabas-bot to a computer.
- Learn procedure that allows code to be uploaded to the robot.
- Learn the function of three blocks in Ardublock: Loop do, Set digital pin, and delay.
- Understand that the robot can be physically manipulated through code.

Independent Practice

- Use Ardublock to turn the LED on, then to turn the LED off.
- Create a program in Ardublock that continuously blinks the light on and off.

Final Assessment, Project, or Product

- Determine the smallest millisecond delay that still appears to blink.

Lesson Materials (per student)

- ☐ 3V LED (x1)
- ☐ Jumper wires
- ☐ Resistors (470 Ohm - yellow violet brown gold and 4.7K Ohm - yellow violet red gold)
- ☐ Mini breadboard (x1)
- ☐ Barnabas Noggin (x1)
- ☐ Computer/laptop (x1)
- ☐ USB to micro USB cord (x1)
- ☐ Engineering Journal (x1)
- ☐ Pencil (x1)



WEEK 9 OF 16: INTRODUCTION TO CODING: THE BLINKING LIGHT CHALLENGE!

Lesson Plan

Review

1. What is resistance? What is the job of a resistor in a circuit?

Step 1: Connecting The Barnabas-Bot (10 minutes)

Rather than go through the process here, please look at the Additional Resources provided in the Lesson overview. There are two separate documents. One for Mac users and one for PC users. There you will find instructions to connect the Barnabas-bot to the computer. They will also include instructions on how to ensure the robot and computer can communicate with one another (upload code).

Vocabulary

- **Arduino:** Both hardware and software. The hardware is a line of microcontrollers (very simple computers). In fact, the Barnabas Noggin utilizes an Arduino microcontroller itself. The software is a C based IDE (integrated development environment... A place where you code). Because Arduino handles both sides of things the process of coding, then uploading your code to a microcontroller is very easy. The Arduino IDE can be out of scope for younger classes as the ability to type acts as a barrier to entry.



Have the students make a step by step checklist on how to connect the Barnabas-Bot to the computer in *explicit detail*. The checklist should look something like the following:

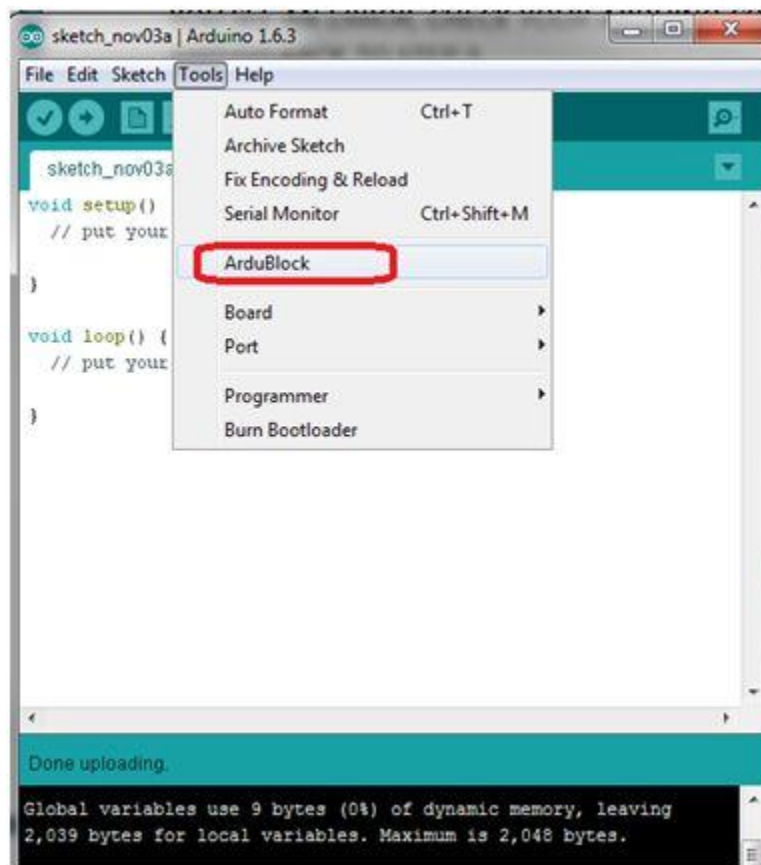
1. Turn on the computer.
2. Input password.
3. Open Arduino.
4. Plug the robot into the computer via the USB cable.
5. Go to tools -> port -> select port (not com1 or serial port ex. for windows pc).
6. Go to tools -> board -> select Arduino Nano

This process must be done each lesson that you intend to program.

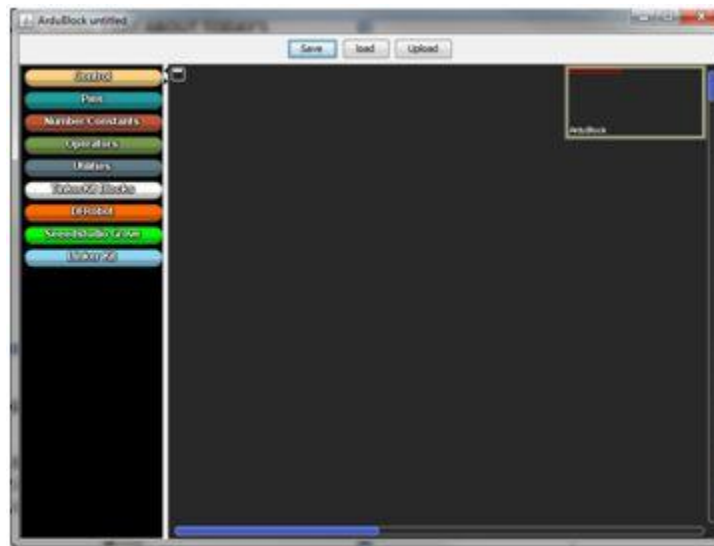
Step 2: Getting Started With Ardublock (5 minutes)

Now that our computer is talking to our Barnabas Noggin, let's get started with ArduBlock! ArduBlock is a block-based programming language that is great for those who are new to programming.

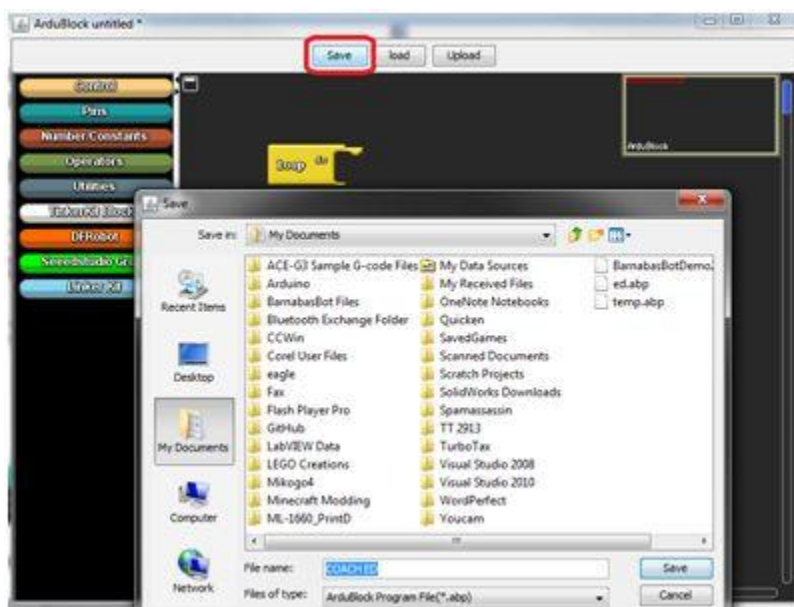
Open ArduBlock by going to **TOOLS->ARDUBLOCK**.



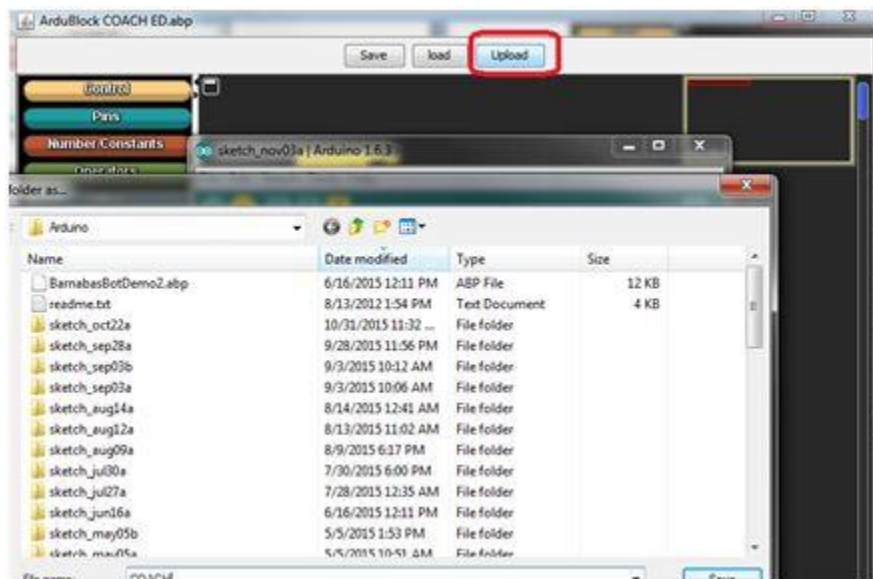
Once ArduBlock opens, you should see the following screen. Note that your screen may differ a little depending on which version of Ardublock you have installed.



Go to [CONTROL], and drag in a [LOOP-DO] block (if it isn't there already). Click on [SAVE], type in your name, and click [SAVE] again. Note that every program needs a [LOOP-DO].



Now click [UPLOAD], type in your name, and click [SAVE]. Your program will now upload code into your robot.



Congratulations, you just wrote your first program! In the future, you will want to always click [Save], and then [Upload] when loading your program to your Barnabas-Bot.

Vocabulary

- Ardublock: A GUI (graphic user interface... another place to code) that runs with Arduino. This means that closing Arduino also closes Ardublock so be careful! Ardublock is a block based coding platform, allowing us to sidestep the need to type.



Uploading blank code may seem frivolous but it is important. By attempting to upload now any technical issues can be addressed. If the Arduino window shows any kind of error, you know there is a problem.

The three most likely reasons a student would get an error message here are: They did not choose the correct port, they did not choose the correct board, or their robot is not connected to the computer.

Step 3: Ardublock Basics (5 minutes)

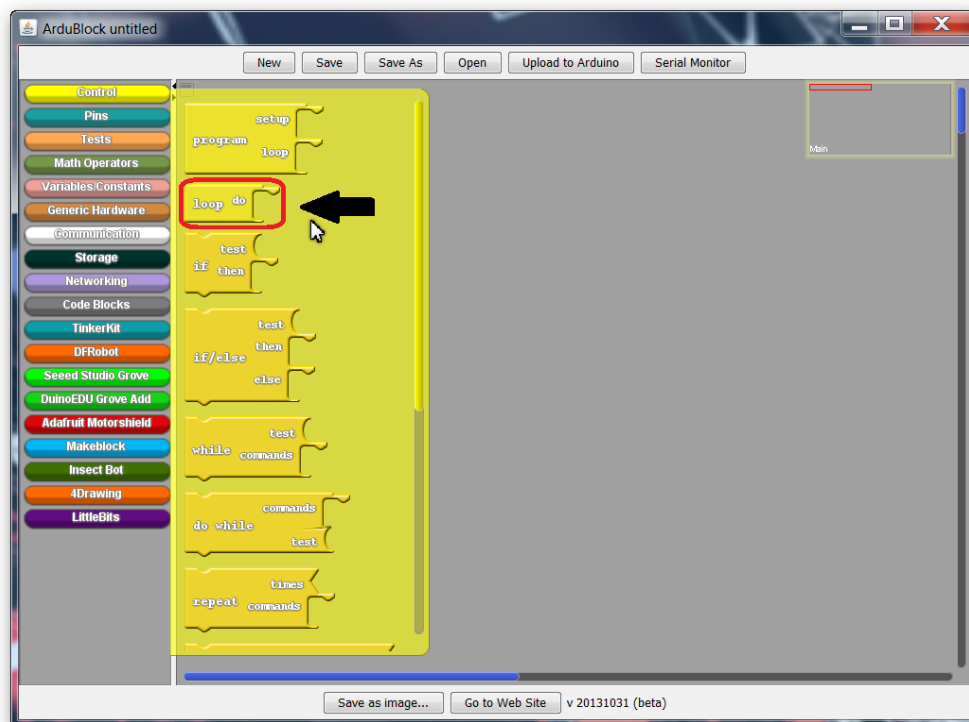
Before we can begin coding we should familiarize ourselves with the basics of ardublock. How to move around, create and delete code.

The first thing to mention is the loop do. The loop do is the most important block in Ardublock. Code is only able to be uploaded to your robot if there is one and only one loop do. The loop

do will house any other blocks used in our code and will change its size to accommodate. As inferred by its name, the loop do will loop through your code. When it reaches the end of your code, it will start again at the beginning. Luckily for us, this integral part of our code will always be in a new Ardublock file by default.

Let's say that the loop do is not there when opening Ardublock. We can add the loop do to the code, as well as any add any other block, by the following method. All of the colored tabs on the left side of the Ardublock window can be selected. After selecting one of them a menu will pop up with several similarly colored blocks. You can drag any of the blocks into the grey coding area (generally putting them inside of the loop do).

In the case of the loop do you would navigate to the yellow tab labelled control, then drag and drop loop do into the gray area.



The last skill we need to learn is how to delete blocks of code. Say we have the case of multiple loop do's, and we want to get rid of the extras. Just use your cursor to grab one of the unwanted blocks and drag it over to the tabs on the left side of the window. Then, just drop it and it should disappear.

Vocabulary

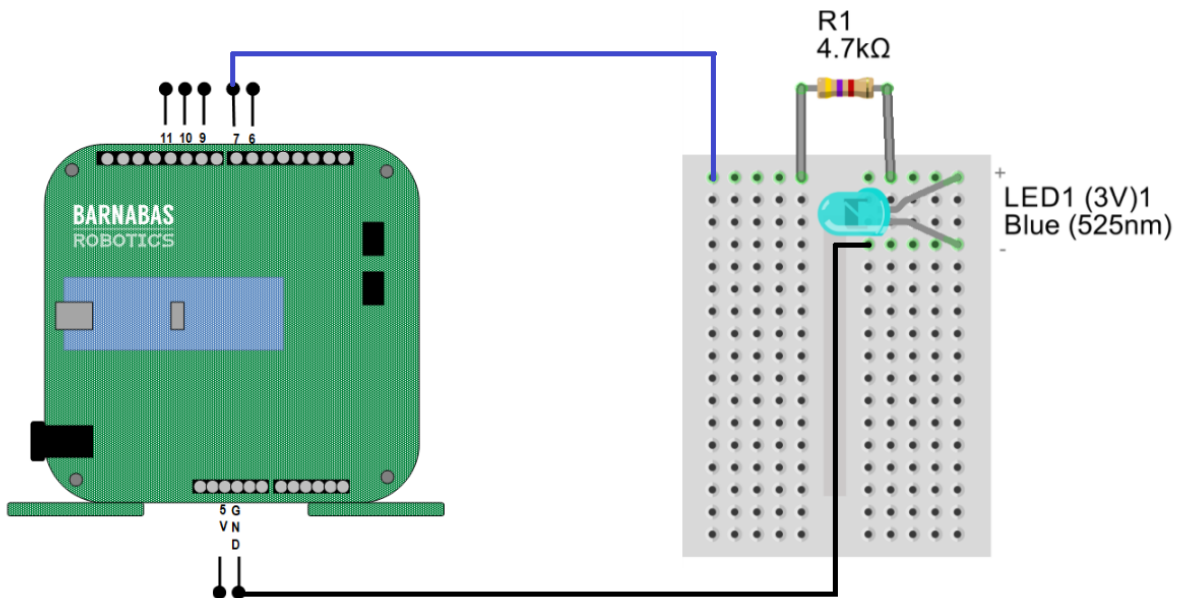
- **Loop Do:** The most important block in Ardublock. It is important that one and only one loop do is in the code at a time, otherwise your code cannot be uploaded. The loop do

will run whatever code is placed inside it repeatedly, starting again at the beginning whenever the end is reached.

Step 4: Our First Program, Programming The LED (20 minutes)

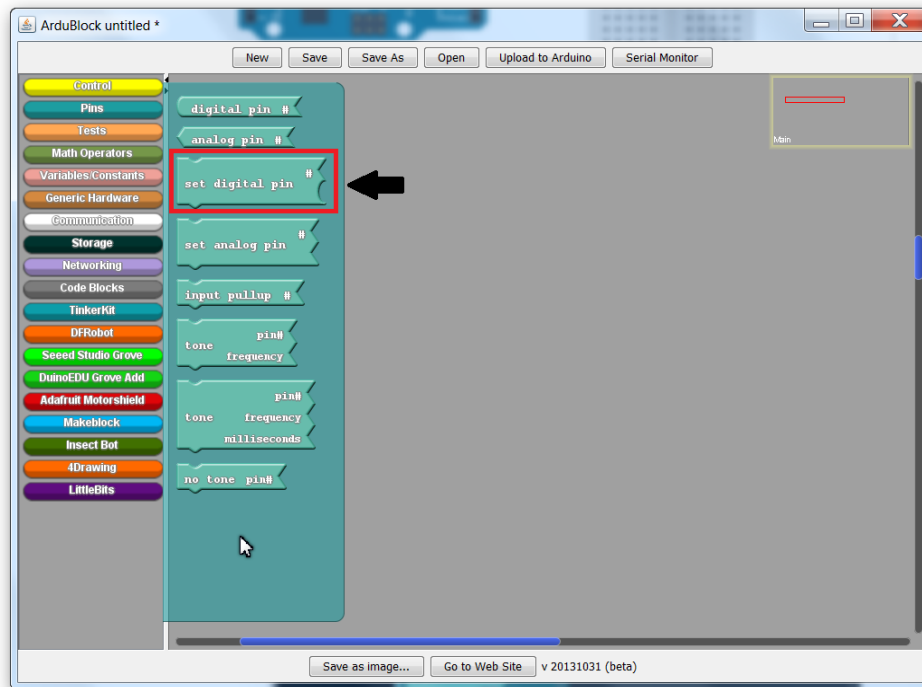
In this section we are finally going to code our LED. However, we must make a small change to the current circuit first.

The wire that is currently in the 5V pin must be placed in the pin labelled 7.

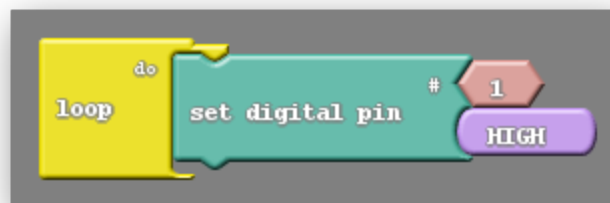


You may notice that the LED no longer turns on. This is because while the 5V pin is always on, pin 7 is what is called a programmable pin, meaning we decide whether it is on or off with code.

With that out of the way let's begin coding by grabbing a Set Digital Pin block out of the Pins tab and place it in the loop do.



The Set Digital Pin block should snap into place when dropped on the loop do.



There are two smaller blocks attached to the Set Digital Pin block. The number 1 and the word HIGH. The number 1 refers to the pin on the robot that this block will attempt to control. Remember that we placed the LED circuit on pin 7, so that 1 needs to be changed to a 7. This can be done by clicking on the block, pressing 7, then pressing enter.

The HIGH block is telling us that it will be turning the pin on (giving it power). By mousing over this block with your mouse cursor and clicking on the upside down triangle that appears on it a menu will appear. That menu has only two options HIGH and LOW. As you may have guessed LOW will turn the pin off.

After changing the pin number to 7 have all the students press upload to arduino at the top of the Ardublock window. You will be prompted to save, go ahead and do so. Afterwards you will see a green progress bar in the Arduino window showing the code being uploaded. You should see the LED turn on.

Next you should ask the students to turn the LED off by changing the code. Let them figure out how to do so on their own (Switch the HIGH to LOW). In order to see the change the students will need to upload their code again. The Barnabas Noggin is only capable of holding one program at a time. This means that uploading the LED off code will permanently erase the LED on code from the robot. To turn the LED on again they would have to re-upload a set digital pin HIGH code to the robot.



LED on



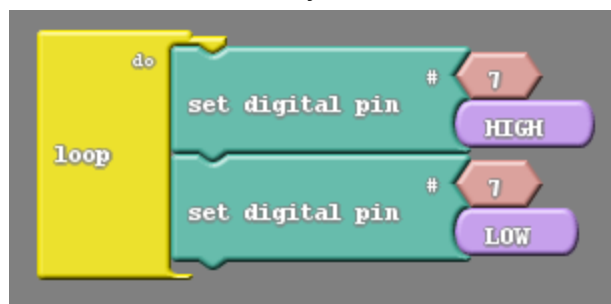
LED off

Vocabulary

- Programmable pin: One of the pins on the Barnabas Noggin labelled 0-13 (it has more labelled A0-A5, but we are not using those in this class). These pins can be controlled by the code we create in Ardublock rather than just being innately on or off.
- Set Digital Pin: Located in the aqua colored pins tab, this is the block we use to turn the LED on and off. To properly use this block we need to pick a pin number (for us 7) and choose between HIGH or LOW (on or off). This block must be placed inside the loop do.

Step 5: Blinking The LED (20 minutes)

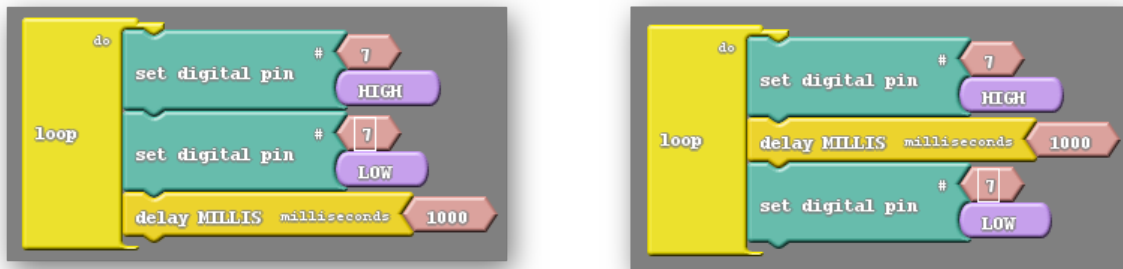
Now it's time for the big challenge, blinking the LED. I generally start by asking the class for suggestions on how to do this. The answer I'm looking for at this point is that we should place another Set Digital Pin block into our loop do. By having two Set Digital Pin blocks, one HIGH and the other LOW, We should see both behaviors happen. Have the students do this and upload the code. What kind of behavior do they see?



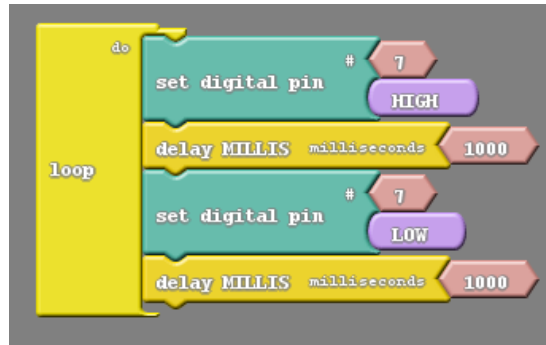
They should see the light turn off momentarily before turning back on, and staying on there after. Don't be fooled by the initial turning off of the LED, that is not do to our code working as intended. The Noggin will always stop whatever it is doing to accept new code. In this case turning the LED off just after our code is uploaded. Unfortunately this is not part of a blink. Because the loop do repeats we should see the LED blink over and over, which we do not with the code as written.

The reason we can not see a repeated blink is due to the speed at which the Barnabas Noggin is processing commands. The Barnabas Noggin thinks fast enough to do hundreds of commands in a second, far to fast for our eyes to keep up with. The LED *is* blinking, but too fast for us to make out.

There is a way to slow our code down deliberately. In the controls tab there are two blocks labelled Delay. One says Delay milliseconds (a millisecond is one thousandth of a second), and the other says delay microseconds (a microsecond is one millionth of a second). We will always use the Delay milliseconds block, as the Delay microseconds block yields the same problem as before; the LED will be blinking too fast for us to see. The question is where do we place the Delay millisecond block in our current code. There are two immediate options:



You're probably wondering why there isn't a third option for the delay block being at the top of the code. It turns out that would be identical to the code on the left due to the code repeating. Unfortunately, these don't work either. The one on the left seems to stay off, and the one on the right seems to stay on. In either case the delay block is only pausing after one of the actions, either turning off or turning on. We need to pause our code after each action, allowing both of them to happen for some observable amount of time, like below:



Vocabulary

- **Delay:** The delay block comes in two different flavors, delay milliseconds and delay microseconds. We only make use of the delay milliseconds block. The delay block will stop code from progressing forward until a certain amount of time has elapsed.



Once your students have the LED blinking challenge them to change the numbers inside the delay block and identify how that changes the behavior of the LED. Challenge the students to find the smallest delay where they can still see the LED blink.



REFLECTION

Comprehension

1. What significance does changing the number attached to the Set Digital Pin block to 7 have?
2. What effect does the delay block have? Why do we only use the delay milliseconds block?
3. Why do we need two delay blocks to make the LED blink?

Challenges

What was hard?

Enjoyment

What was fun?

Mindset

What did you learn about yourself? What do you need to work on?

Community

How can what you learned impact those around you?



WEEK 10 OF 16: MORSE CODE WITH THE LED

Suggested Time: 60-75 minutes

Lesson Overview

Disciplinary Core Ideas

Next Generation Science Standards

- ETS1.B: Developing Possible Solutions: Tests are often designed to identify failure points or difficulties, which suggest the elements of the design that need to be improved. (3-5-ETS1-3)
- At whatever stage, communicating with peers about proposed solutions is an important part of the design process, and shared ideas can lead to improved designs. (3-5-ETS1-2)
- Science is a Human Endeavor: Most scientists and engineers work in teams. (4-PS3-4)

Learning Target(s)

Technical Skills

- Knowledge of how morse code functions.
- Understanding of how to condense code.
- Improved understanding of how code is executed block by block.

Life Skills

- Planing
- Testing
- Altering a hypothesis

Essential Questions

1. What is morse code? How does one communicate using morse code?
2. How can we condense our code to make blinking in morse code easier to achieve.
3. Does your code vary the time the LED spends off? Why or why not?

Key Vocabulary

1. Morse code
2. Repeat block

Additional Resources

Videos

- None

Documents

- None

Other

- [Morse code PDF](#)

Depth of Knowledge Levels Addressed

Level 1: Recall and Reproduction

Level 2: Skills and Concepts

Level 3: Strategic Thinking and Reasoning

Level 4: Extended Thinking

Barriers To Learning

- Poor understanding of the previous lesson.

Anticipatory Set

- Learn morse code and apply that knowledge to create a message in morse code with the LED.

Independent Practice

- Programming in an attempt to create different letters in morse code by causing the LED to blink at different intervals.

Final Assessment, Project, or Product

- Create an S-O-S message in morse code using the LED.
- Create customized messages, such as their names, in morse code with the LED.

Lesson Materials (per student)

- ☐ Barnabas Noggin (x1)
- ☐ USB to micro USB cord (x1)
- ☐ 3V LED (x1)
- ☐ Jumper wires
- ☐ Resistors (470 Ohm - yellow violet brown gold and 4.7K Ohm - yellow violet red gold)
- ☐ Mini breadboard (x1)
- ☐ Computer (x1)
- ☐ Engineering Journal (x1)
- ☐ Pencil (x1)



WEEK 10 OF 16: MORSE CODE WITH THE LED

Lesson Plan

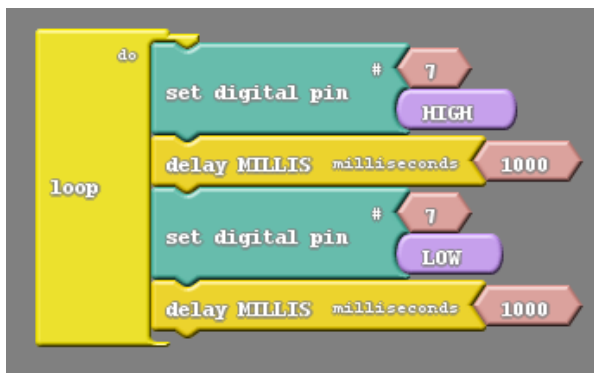
Review

1. What block is used to turn the LED on and off.
2. How is the delay block important in getting the LED to visibly blink?

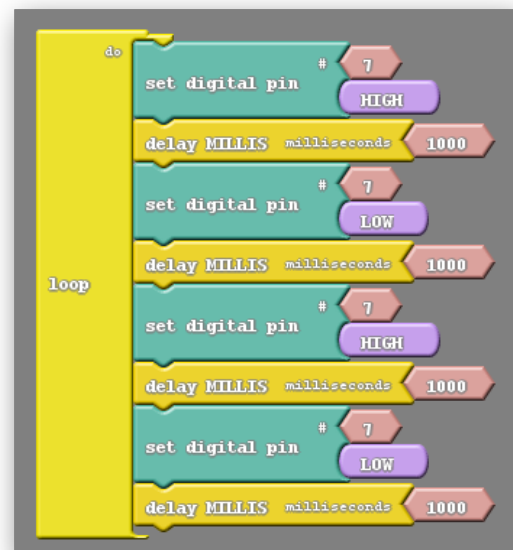
Step 1: Blinking At Different Speeds (25 minutes)

We spent last week learning to blink our LEDs and playing with the speed at which it blinks. We are going to pick up where we left off by using the skills learned last week. The first challenge of this lesson is to create a program that causes the LED to blink at two distinct speeds. It will be on for a long period of time (long as in about a second), then off, then on for a short period of time, then off.

How should our code from last week, the blinking light code, be altered to accomplish this? Well we need to first understand we will need two different blinks, and will therefore need more blocks. If we stick to the status quo of four blocks in our code we will be limited to only one of the two behaviors; either the fast blink or the slow blink.



One blink



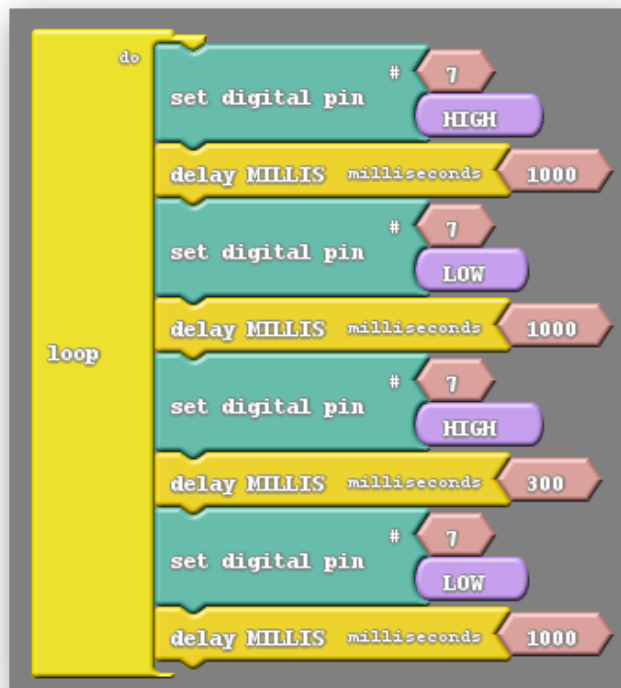
Two blinks

In the example on the right we have enough blocks to create two distinctly different blinks, however at the moment the code on the right and the code on the left behave identically. Why is that?

The right hand code may have enough blocks for two blinks but currently all of the delay blocks have the default value of 1000. So in either code every blink we see will be the same length of time. In order to vary the blink speeds we need to alter the value of one or more of the delay blocks. The question is which delays need to be altered? Let's have a look at the code below:



You can see here that the first two delays are kept at the same value, and the last two delays are kept at the same value. They are separated into two blinks, each with their own specific delay. There is a small problem with this code however, which will be apparent as you watch the LED. The second blink, in particular the low part of it, is short enough that the blink becomes hard to distinguish from the longer blink. This can be solved by making sure that the delays after set digital pin LOW blocks are equal in value, like in the example below:

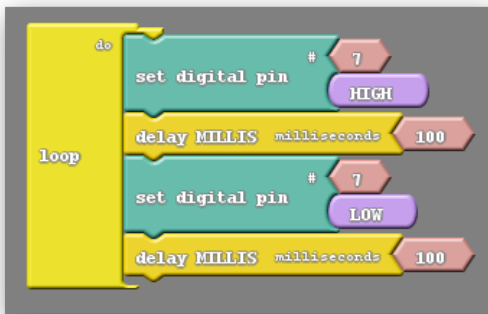


With the delays set at these values both blinks will be clearly visible as separate blinks and it will be clear that one is longer than the other.

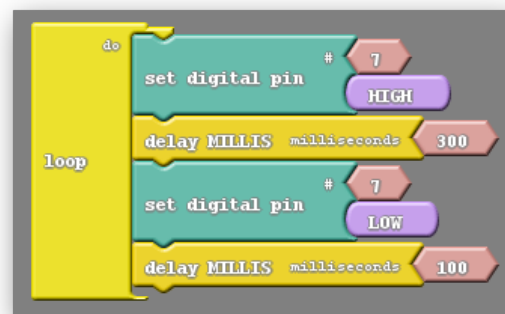
Step 2: What Is Morse Code (10 minutes)

In order to accomplish this lesson's challenge we first need to understand morse code. Both what morse code is as well as how it is used. Morse code is essentially an alternate way of producing the alphabet using an assortment of dots and dashes to represent any one letter. A message in morse code is usually conveyed through either light or sound rather than just on paper. At this point I'd like you to look at the morse code PDF [located here](#) as well as in the additional resources section at the top of the lesson. This is a resource that you should consider distributing to your students.

That PDF does ask that the short and long blinks be timed in a different manner. It asks for significantly shorter blinks in both cases, which would look like the following:



Short blink. A dot.



Long blink. A dash.

In addition the morse code pdf gives us rules for ending letters and ending words. We are told to have a 1000 millisecond delay at the end of a letter and a 2000 millisecond delay at the end of a word. This helps us interpret the morse code message a little easier, as we can better distinguish between individual letters and words.

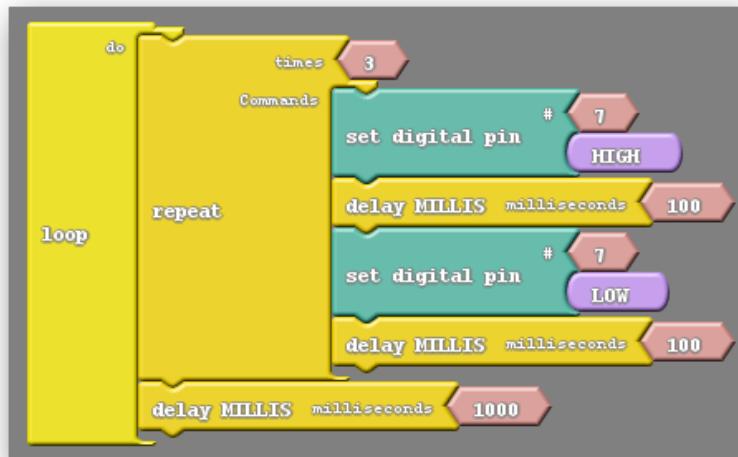
Vocabulary

- **Morse Code:** A method of transmitting information as a series of on-off tones, lights, or clicks that can be interpreted by a skilled observer without the use of equipment.

Step 3: Our First Morse Code Message (20 minutes)

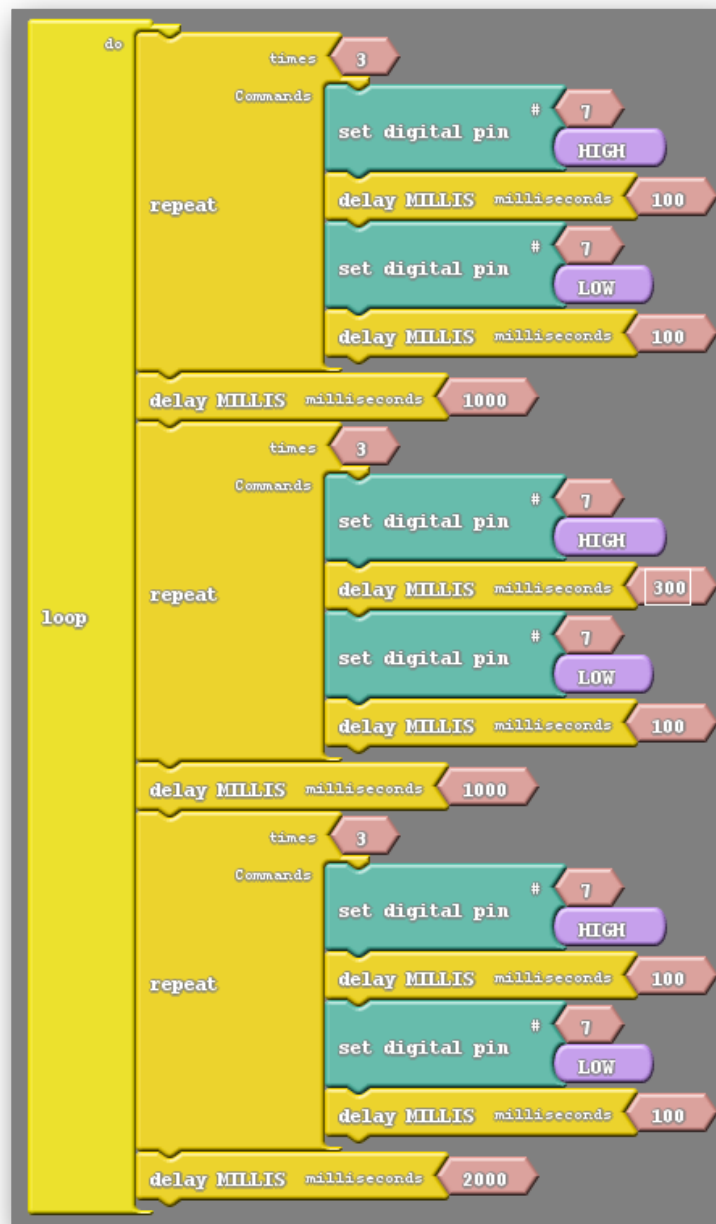
Morse code is probably most commonly known as an emergency method of communication, with the most commonly known message being S-O-S (help). This will be the first message we try to create using the LED. It is actually one of the easiest messages to create in morse code as S is represented by three dots, and O is represented by three dashes. Because both S and O are defined with repetitive signals it would be best for me to introduce a new block.

The repeat block can be found in the control tab. It is the first of the three repeat blocks seen, the others being repeat and count, and repeat between. This block behaves somewhat like the loop do in that other blocks can be placed inside it. There is a small integer block attached to it, much like the other blocks we have used thus far, which can have its value changed. That number will dictate how many times the code within the repeat block will run before further blocks in your code are. Take a look at the example below:



This code will blink fast three times before waiting a second, then starting over. Based on what we already know, this code creates the message S-S-S-S-S.....

Putting together the rest of the S-O-S code should be easy to accomplish. We just need to add three long blinks, followed by three more short blinks, with the appropriate delays in between.



Notice that the last delay is for 2000 milliseconds (2 seconds). This is appropriate if we consider S-O-S to be a word.

Vocabulary

- **Repeat:** The repeat block, like the loop block, can have other blocks placed within it. Any blocks within the repeat block will be run a number of times equal to the integer that innately comes with the block. Afterwards, subsequent blocks of the code will be run as expected.



Have the students attempt custom messages in morse code, such as their name, and see if other students can decipher them.



REFLECTION

Comprehension

1. What is morse code? How is morse code communicated?
2. When making blinks of different speeds, how does not changing the delay after turning the LED off benefit us?
3. Describe how the repeat block helps us create code?

Challenges

What was hard?

Enjoyment

What was fun?

Mindset

What did you learn about yourself? What do you need to work on?

Community

How can what you learned impact those around you?



WEEK 11 OF 16: ASSEMBLING THE ROBOT

Suggested Time: 60-75 minutes

Lesson Overview

Disciplinary Core Ideas

Next Generation Science Standards

- ETS1.B: Developing Possible Solutions: Tests are often designed to identify failure points or difficulties, which suggest the elements of the design that need to be improved. (3-5-ETS1-3)
- At whatever stage, communicating with peers about proposed solutions is an important part of the design process, and shared ideas can lead to improved designs. (3-5-ETS1-2)
- Science is a Human Endeavor: Most scientists and engineers work in teams. (4-PS3-4)

Learning Target(s)

Technical Skills

- Fine motor skills
- Following a procedure

Life Skills

- Perseverance
- Teamwork

Essential Questions

1. None

Key Vocabulary

1. Fastener

Additional Resources

Videos

- [Motor and arm assembly](#)

Documents

Other

- Appendix C (reference if constructing wood bot version)

Depth of Knowledge Levels Addressed

Level 1: Recall and Reproduction

Level 2: Skills and Concepts

Level 3: Strategic Thinking and Reasoning

Level 4: Extended Thinking

Barriers To Learning

- There should be no significant barriers in this lesson

Anticipatory Set

- Students will assemble their robots.

Independent Practice

- Following step by step instructions to assemble the robot.

Final Assessment, Project, or Product

- Fully assembled robot.

Lesson Materials (per student)

- ☐ 3 x Servo Motors (9g)
- ☐ Screw Bag A:
 - ☐ 8 x #2-9/16" Screws
 - ☐ 8 x 2-56 Nuts
 - ☐ 1 x Miniature Phillips Screwdriver
- ☐ 1 x Robot body
- ☐ 2 x Robot arm
- ☐ 1 x Robot front plate
- ☐ 1 x Barnabas noggin
- ☐ 1 x Robot head (top + bottom)
- ☐ Screw Bag B:
 - ☐ 6 x 1/4" screws



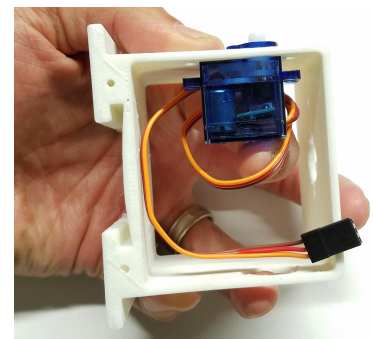
WEEK 11 OF 16: ASSEMBLING THE ROBOT

Lesson Plan (See Appendix C If Constructing Wood Bot Version)

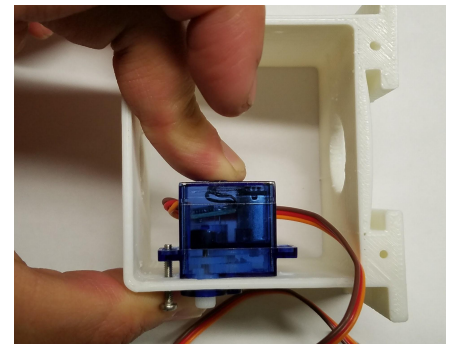
Step 1: Attaching The Servo Motors

Servos motors are used to move your robot's right arm, left arm, and head. Servo motors can move a half-circle (or 180 degrees). Follow the below steps to attach the servo motors:

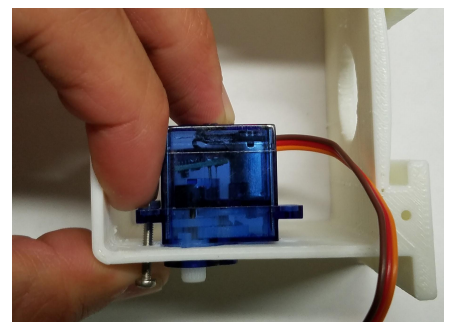
- Take one servo motor and place it into the left arm socket of the robot body. Note: The servo motor should be placed inside of the plastic robot body casing. The top of the motor (with the white plastic tip) will stick slightly out of the robot's body. (See photo.)



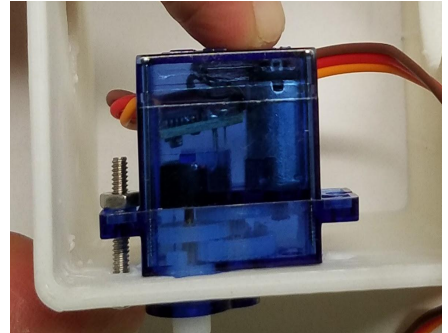
- Take one 9/16" servo screw and put it through the outside of the body and into the corresponding hole on the wing of the servo motor. Push it through so that you can see the screw stick out through the wing of the servo motor. If you have trouble sticking the screw through, try using the screwdriver to screw it in.



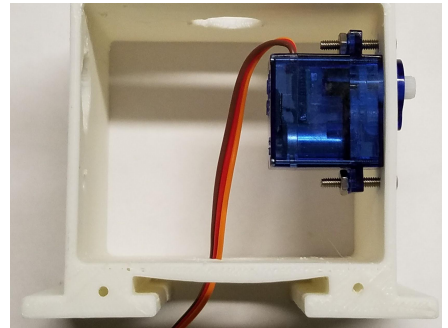
- Take a nut and place it over the tip of the 9/16" servo screw. Then, while gently holding the nut in place with your finger, tighten the 9/16" servo screw using your screwdriver. (Be careful not to over tighten.) (This may require some patience and perseverance. But you can do it!)



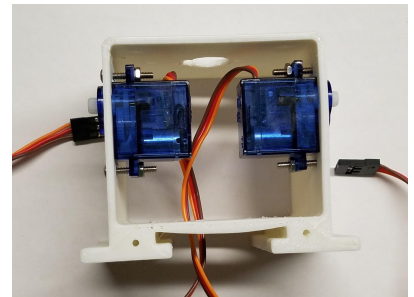
- Once your nut is secure on your servo screw, it should look like the photo.



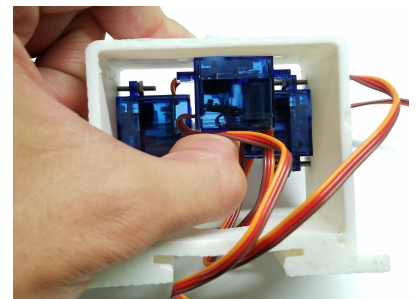
- Repeat the above instructions for the other wing of the servo motor. Once completed, your servo motor should look like the photo.



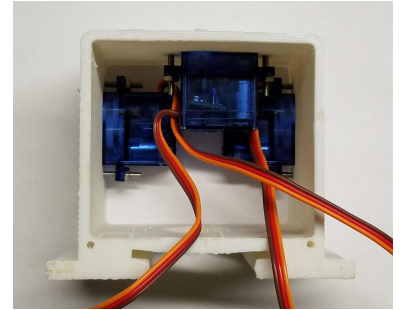
- Now, repeat the above steps for the other arm hole. Once you have finished, your robot's body should look like the photo below.



- Now that your two arm motors are assembled, take the last servo motor and fit it into the third servo motor socket on the top of the robot body. This third motor will be for your robot's head. Repeat the steps above to attach this third servo motor to your robot body.



- Once completed, your robot should look like this.



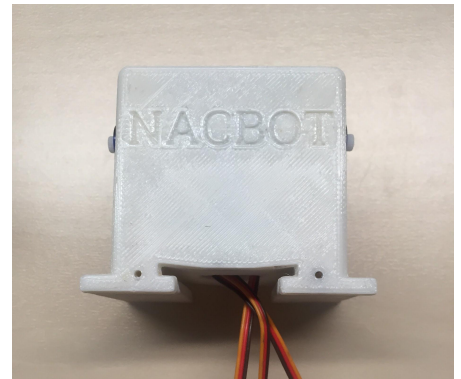
Vocabulary

- Fastener: a hardware device that mechanically joins or affixes two or more objects together.

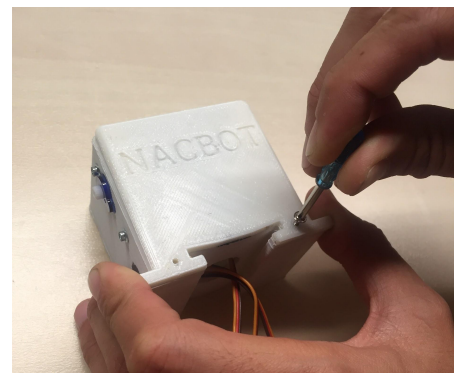
Step 2: Frontplate Assembly

In this step, you will now cover your robot's motors by installing your faceplate by following the steps below:

- Find the front of your robot body. (Hint: It is the open side of the body that is closest to the arm motors and has 2 holes at the bottom.) Align your faceplate to your robot's body so the screw holes line up. (See photo.)



- Using your miniature Phillips screwdriver, fasten the two 1/4" screws to the robot's body.



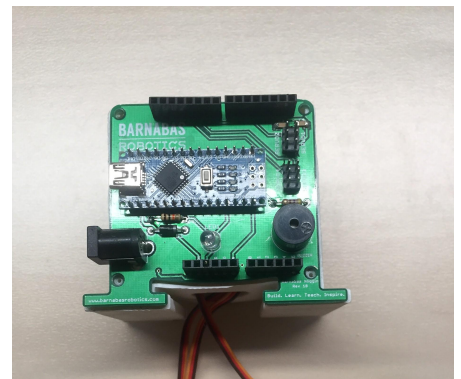
- Once completed, your robot should look like the photo.



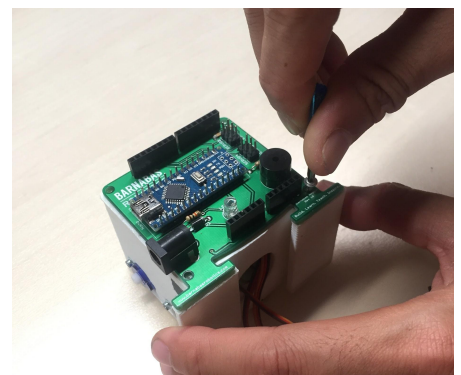
Step 3: Attaching The Backplate

In this step, you will now cover the backside of your robot's body by installing your Barnabas Noggin. Follow the step below:

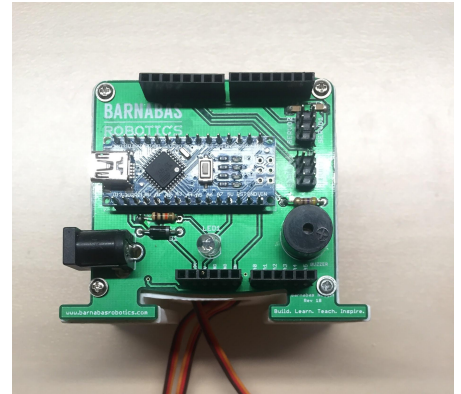
- Find the back of your robot. (Hint: It is the open side of the body that is furthest from the arm motors with 4 holes.) Align your Barnabas Noggin to your robot's body. The screw holes on the four corners of the robot body and the Barnabas Noggin should line up. (See photo.)



- Next, screw in the four 1/4" screws into the four holes using your screwdriver.



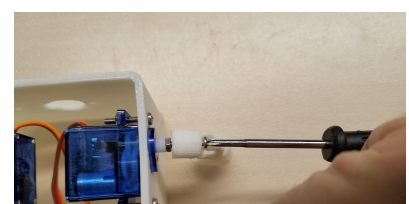
- Once completed, your robot should look like the photo.



Step 4: Attaching The Arms

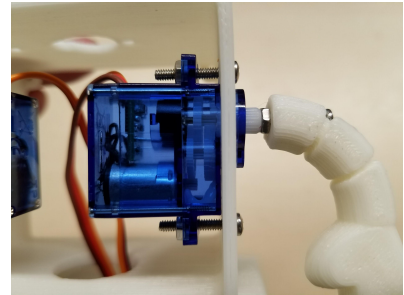
Now you will connect your arms to your robot's body by following the steps below:

- Take one 9/16" servo screw and put it through one of the arm's holes. Screw on one nut to the other side of the servo screw--but only part of the way. Be careful not to screw the nut in all the way. (See photo.)
- Take the arm and hold it up to the servo motor shaft (i.e. white plastic tip of the servo motor) as shown below. The hole in the arm should match up with the hole on the servo motor shaft.

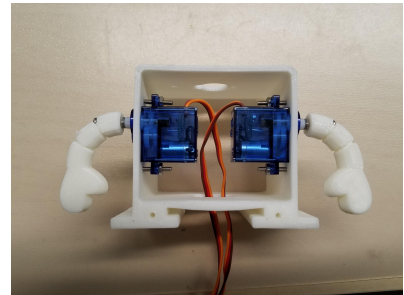


- Next, using the screwdriver, tighten your servo screw by turning clockwise. (Be careful not to over tighten.) (See photo.)

- Once the arm is tightened, it should look like the photo.



- Now, repeat the process above for the other arm too. Once you are finished, your robot arms should look like the photo.



Step 5: Attaching The Head

Every robot needs a head! You will be fastening the head to the servo motor sticking out of the top of the robot as described below:

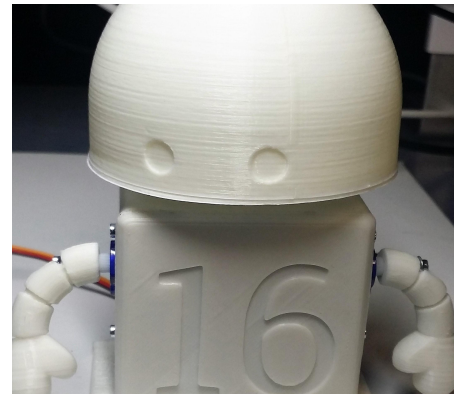
- Find the bottom of your head (i.e. the base) and align the hole to the top of the hole on the servo motor shaft head. (See photo.)



- Fasten the screw.



- Now fasten the top of the head to the bottom part. It should fit in over the top. You may want to use glue (Elmer's glue or super glue) to secure it further.



The typical message of perseverance is particularly relevant this lesson as the need for fine motor skills often frustrates students. This is something as a teacher to be aware of.



REFLECTION

Comprehension

1. Why is fastening the motors from both sides important?

Challenges

What was hard?

Enjoyment

What was fun?

Mindset

What did you learn about yourself? What do you need to work on?

Community

How can what you learned impact those around you?



WEEK 12 OF 16: THE BUZZER

Suggested Time: 60-75 minutes

Lesson Overview

Disciplinary Core Ideas

Next Generation Science Standards

- Planning and Carrying Out Investigations: Plan and conduct an investigation collaboratively to produce data to serve as the basis for evidence, using fair tests in which variables are controlled and the number of trials considered. (3-5-ETS1-3)
- ETS1.C: Optimizing the Design Solution: Different solutions need to be tested in order to determine which of them best solves the problem, given the criteria and the constraints. (3-5-ETS1-3)

Learning Target(s)

Technical Skills

- Deeper understanding of Arduino hardware, I/O pins.
- Use previous coding knowledge to sound buzzer at different frequencies and make music.

Life Skills

- Communal responsibility

Essential Questions

1. What relationships can be found between patterns and music?

Key Vocabulary

1. Tone
2. No Tone

Additional Resources

Videos

- [Building the buzzer circuit tutorial](#)
- [Coding the buzzer tutorial](#)

Documents

- None

Other

- [Frequency chart](#)

Depth of Knowledge Levels Addressed

Level 1: Recall and Reproduction

Level 2: Skills and Concepts

Level 3: Strategic Thinking and Reasoning

Level 4: Extended Thinking

Barriers To Learning

- Poor understanding of blinking light lesson
- Inability to distinguish between known blocks in Ardublock

Anticipatory Set

- Students will learn to code their buzzers at different frequencies to create different tones

Independent Practice

- Creating the buzzer circuit
- Experimenting with the different tones the buzzer can make through coding

Final Assessment, Project, or Product

- Creating a coherent array of tones such as a siren, a scale or a song

Lesson Materials (per student)

- ☐ 1 x Buzzer
- ☐ 1 x Resistor (4.7k Ohms)
- ☐ 2 x Jumper wires
- ☐ 1 x Engineering journal
- ☐ 1 x Pencil
- ☐ 1 x Computer with Arduino/Ardublock software
- ☐ 1 x Assembled robot

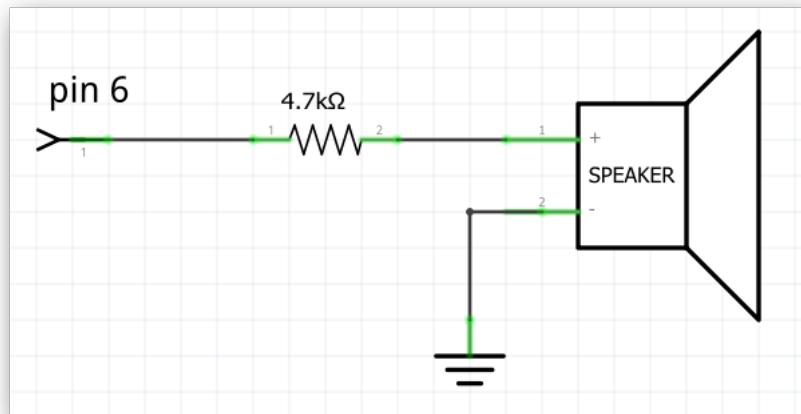


WEEK 12 OF 16: THE BUZZER

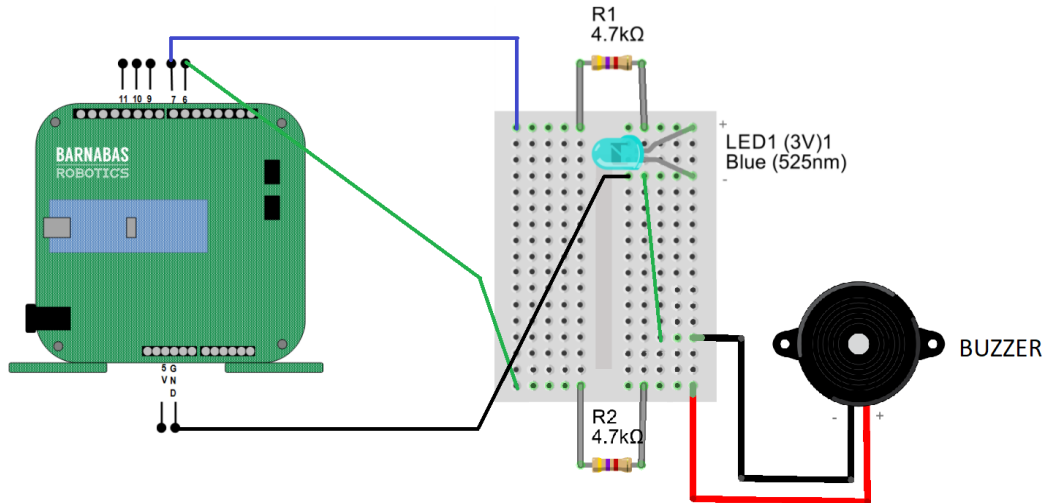
Lesson Plan

Step 1: Building The Buzzer Circuit (10 minutes)

In this lesson we will be adding the buzzer to our robot in the same way we added the LED. There is no need to take the LED circuit off the breadboard. We are not replacing it, we are adding to it. Start by drawing the circuit schematic on the board:



In addition to giving the students a schematic to follow have them closely examine the buzzer. They should recognize that one leg of the buzzer is longer than the other, much like the LED. The different length legs, much like the LED indicate that one leg must be placed closer to the positive end of the circuit (the long leg), the high voltage end of the circuit if you will. With this information the students should be able to build the buzzer circuit. It would be wise to construct it a fair distance away from the LED circuit on the breadboard to avoid confusion. If this is not enough then as a class attempt to create a diagram of the circuit on the whiteboard, which would look like the following:



This diagram has the LED circuit on it as well, but that is not necessary unless you intend to have GND common like in the diagram above. What I mean by that is the green wire that is going between the buzzer circuit and the LED circuit is connecting the negative legs (the short ones) of both components. That way, if one is connected to GND, both are connected to GND.



If it is necessary to draw the diagram on the board try having the students participate. Have students suggest additions to the diagram and engage in a conversation about how the circuit needs to be put together

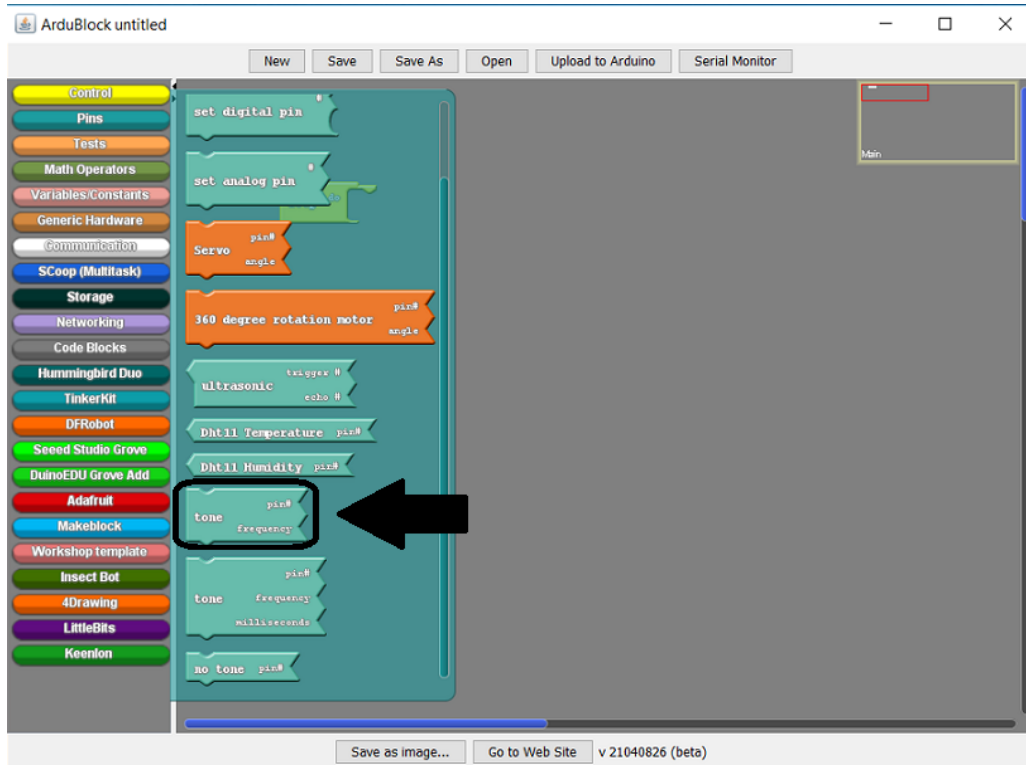
Step 2: How Does The Buzzer Work (5 minutes)

The choice of putting the buzzer on pin 6 is not arbitrary. Pin 6 is what is called a PWM pin, which stands for pulse width modulation. PWM is a fairly complicated process to describe and is beyond the scope of this class. However we can talk about how it functions. A pin that is capable of PWM can vary the voltage that comes from it, as opposed to the other pins which can only give 5 Volts or 0 Volts.

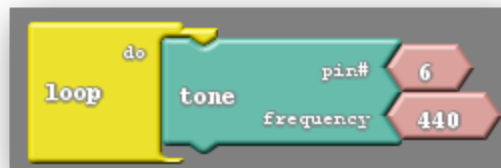
The buzzer will utilize the different voltages by emitting different sounds, different frequencies, depending on the voltage that is currently powering it.

Step 3: Making Tones With The Buzzer (45 minutes)

To program our buzzer we will use the tone block found in the pins tab in Ardublock:



The tone block, like the set digital pin block, requires us to input two pieces of information. Both the pin the buzzer is located on (pin 6), and the frequency of the tone we would like the buzzer to make:



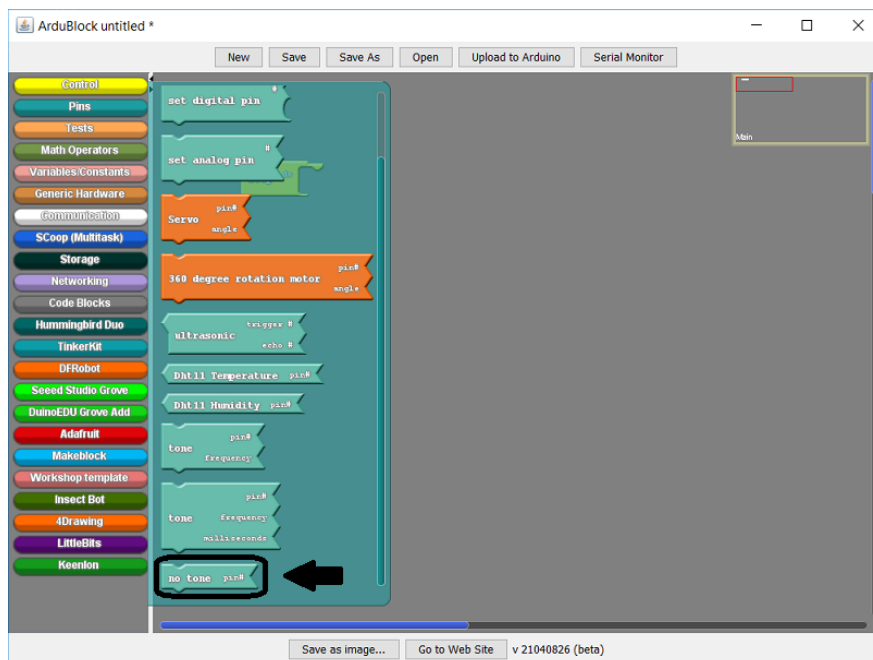
The frequencies that can be chosen relate to different musical note like those shown below:

Musical Notes / Frequencies

Note	Frequency (Hz)	Note	Frequency (Hz)
C ₄	261	C ₅	523
C [#] ₄ /D ^b ₄	277	C [#] ₅ /D ^b ₅	554
D ₄	293	D ₅	587
D [#] ₄ /E ^b ₄	311	D [#] ₅ /E ^b ₅	622
E ₄	329	E ₅	659
F ₄	349	F ₅	698
F [#] ₄ /G ^b ₄	369	F [#] ₅ /G ^b ₅	739
G ₄	392	G ₅	783
G [#] ₄ /A ^b ₄	415	G [#] ₅ /A ^b ₅	830
A ₄	440	A ₅	880
A [#] ₄ /B ^b ₄	466	A [#] ₅ /B ^b ₅	932
B ₄	493	B ₅	987

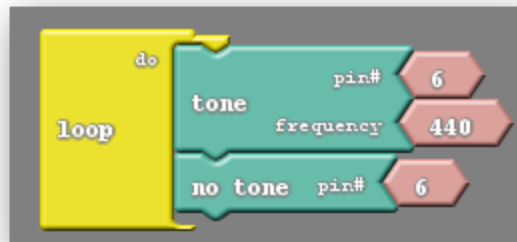
Each of the frequencies shown will make the corresponding musical note. Frequencies that are not seen above can be used, but are generally much less functional. Have your students experiment with different frequencies and share any interesting results they may find.

Once the students have had 2-5 minutes to experience with the tone block you should introduce the no tone block:



The no tone block silences the buzzer. However, much like the LED, we need to be clever in how we code the buzzer so that the buzzer actually behaves the way we want it to.

Let us say that we are trying to have the buzzer make a tone for a small amount of time, then be silent for a similar amount of time. A common mistake is to just put the tone and no tone blocks into the loop do with nothing else accompanying them:



I'm not sure how I would describe the noise the above code makes, but it is definitely not what we are looking for. We have the same issue here that we had when first attempting the blinking light code. Neither of these blocks are given time to take place. Delays need to be added so each block has an opportunity to influence the buzzer:



Instead of the no tone block, we can just place another tone block of a different frequency to make a siren:



Vocabulary

- **Tone:** The tone block is capable of making the buzzer emit sounds of different frequencies. To do so the tone block must be given the correct pin number (for us 6) as well as a frequency. The frequency given dictates how high or low the pitch of the sound is and can be related back to musical notes.
- **No Tone:** The no tone block silences the the buzzer, assuming the correct pin (6) is specified. It does so simply by setting the pin LOW.



Did you know that the human ear can hear frequencies from 20Hz to 20,000Hz. Unfortunately the buzzer isn't capable of making all those sounds. It has an effective range of about 200Hz to 1,000Hz. The buzzer will make noise if you input a

frequency outside of that range, but do not expect the sound it makes to match the frequency given.



The buzzer can be a force for good or for evil. It is often the case that the sound of several buzzers will be overwhelming. This of course depends on the number of students in your class as well as the physical size of your class. Using the 4.7K Ohm resistors like suggested should

mitigate this. You should also keep an ear out for clever students that bypass their resistors altogether, which will create an unmistakably loud tone. Typically I will confiscate buzzers if this occurs.



It is not uncommon to not hear sound after uploading the first time. There are a few common errors that students make when attempting to use the buzzers. Your checklist of things to look for is the following:

- The general mistakes that prevent code from properly uploading. Also the robot being unpowered. In addition the code may be referencing an incorrect pin.
- Put your ear to the buzzer. It may in fact be making noise. The 4.7k Ohm resistor reduces the volume tremendously.
- The circuit may be wired incorrectly. Most often the mistake is that wires are not on the same rows as the legs of the buzzer. The shape of the buzzer can make it difficult to tell which rows the buzzer is sitting on.



The students are capable of doing more than just creating simple sirens using the buzzer. They can create a musical scale easily by following the frequency chart. Some of the more ambitious or musically inclined

students can attempt to recreate common songs. Perhaps twinkle twinkle little star, or mary had a little lamb. The internet is a useful tool for this activity.



REFLECTION

Comprehension

1. How would changing the resistor change the sound coming out of the buzzer?
2. Does changing the frequency affect the volume, pitch, or both?

Challenges

What was hard?

Enjoyment

What was fun?

Mindset

What did you learn about yourself? What do you need to work on?

Community

How can what you learned impact those around you?



WEEK 13 OF 16: INTRO TO SERVOS

Suggested Time: 60-75 minutes

Lesson Overview

Disciplinary Core Ideas

Next Generation Science Standards

- ETS1.B: Developing Possible Solutions: Tests are often designed to identify failure points or difficulties, which suggest the elements of the design that need to be improved. (3-5-ETS1-3)

Learning Target(s)

Technical Skills

- Understanding physical limitations; range of motion (angles), power consumption.

Life Skills

- Perseverance

Essential Questions

1. Do we program the servo motors by giving them angles of position, or angles of motion? What is the difference between the two?

Key Vocabulary

1. Servo motor
2. Servo (block)

Additional Resources

Videos

- [How servo motors work](#)
- [Wiring one servo motor](#)
- [Programming one servo motor](#)

Documents

Other

Depth of Knowledge Levels Addressed

Level 1: Recall and Reproduction

Level 2: Skills and Concepts

Level 3: Strategic Thinking and Reasoning

Barriers To Learning

- Poor understanding of range of motion, angle position vs. angle of motion

Level 4: Extended Thinking

- Inability to distinguish between known blocks in Ardublock
- Poor understanding of previous coding exercises i.e. blinking light and buzzer

Anticipatory Set

- Students will learn to code their motors to move the extremities of the robot and achieve repeated motion

Independent Practice

- Wiring a single motor using the breadboard
- Using code to experiment with the range of motion of the motors as well as the necessary sequence for repeated motion to occur

Final Assessment, Project, or Product

- Getting the robot to wave or shake its head

Lesson Materials (per student)

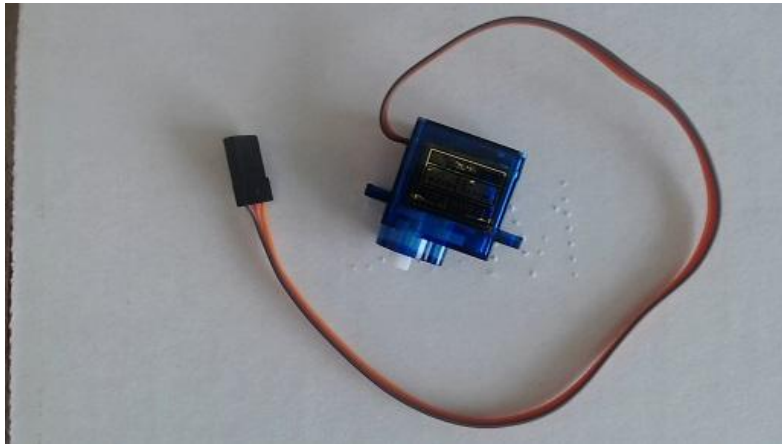
- ☐ 3 x Jumper wires (preferably yellow, red, black)
- ☐ 1 x Engineering journal
- ☐ 1 x Pencil
- ☐ 1 x Computer with Arduino/Ardublock software
- ☐ 1 x Assembled robot



WEEK 13 OF 16: INTRO TO SERVOS

Lesson Plan

Step 1: How Does The Servo Motor Work? (5 min)



Above is a picture of a servo motor. There are multiple types of servo motor, however the important characteristic of our servo motors is that they can only move in half a circle, 180 degrees.

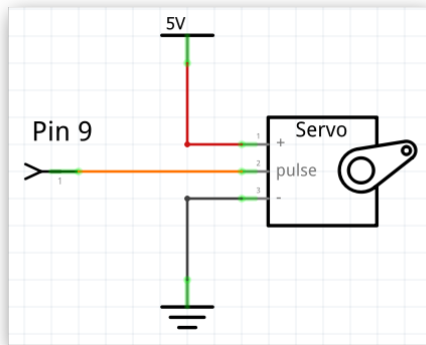
Each servo motor has three wires attached to it: an orange wire, a red wire and a brown wire. If you consider the previous components, the LED and buzzer, you may remember that each of them only needed two wires to function. The main difference between the motor and our previous components is how much current the motor requires. The motor constantly needs 5V fed to it in order to function, where the buzzer and LED only needed 5V intermittently. The motor has two wires dedicated to pumping current in and out of the motor, these are the red (5V) and brown (GND) wires. In our robot human analogy these wires would be the veins. The orange wire, on the other hand, is responsible for sending a signal to the motor which will tell the motor to move. This wire would be more analogous to the nerves that extend across our bodies, carrying signals from our brain to the muscles.

Vocabulary

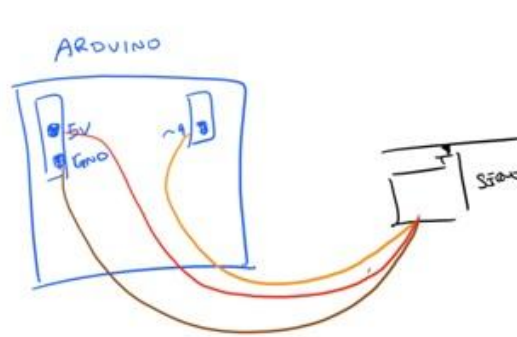
- **Servo Motor:** The servo motors our robot uses are 180 degree rotation servo motors, meaning they only have a range of motion of about half a circle. The servo motors require much more power than the LED or the buzzer, which necessitates a third wire attached to the motor. The motor has one wire dedicated to ground (GND), one dedicated to power (5V), and one dedicated to receiving a signal from the Arduino.

Step 2: Connecting The Motor (10min)

Start by drawing the circuit schematic below:



A more official looking schematic, however it may be difficult for your students to understand what this schematic is saying.

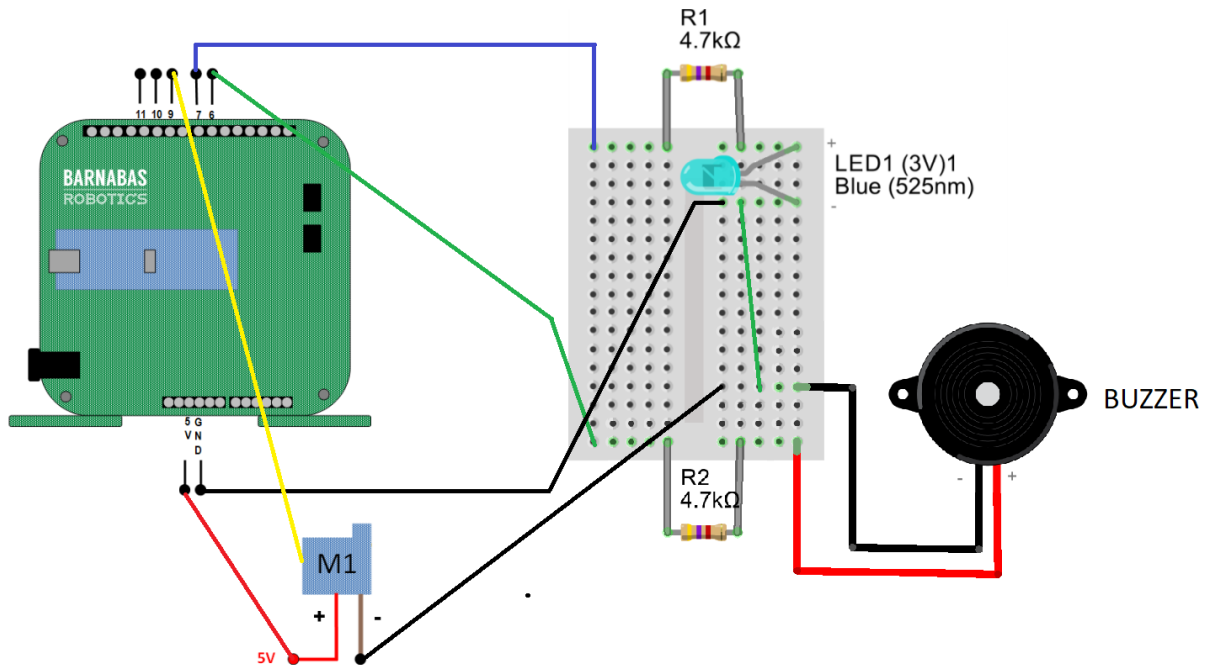


While this schematic looks more primitive, it will probably seem less abstract to the students.

In general I would advise drawing the schematic on the right, however if you are looking to challenge your students you can draw the more abstract schematic on the left.

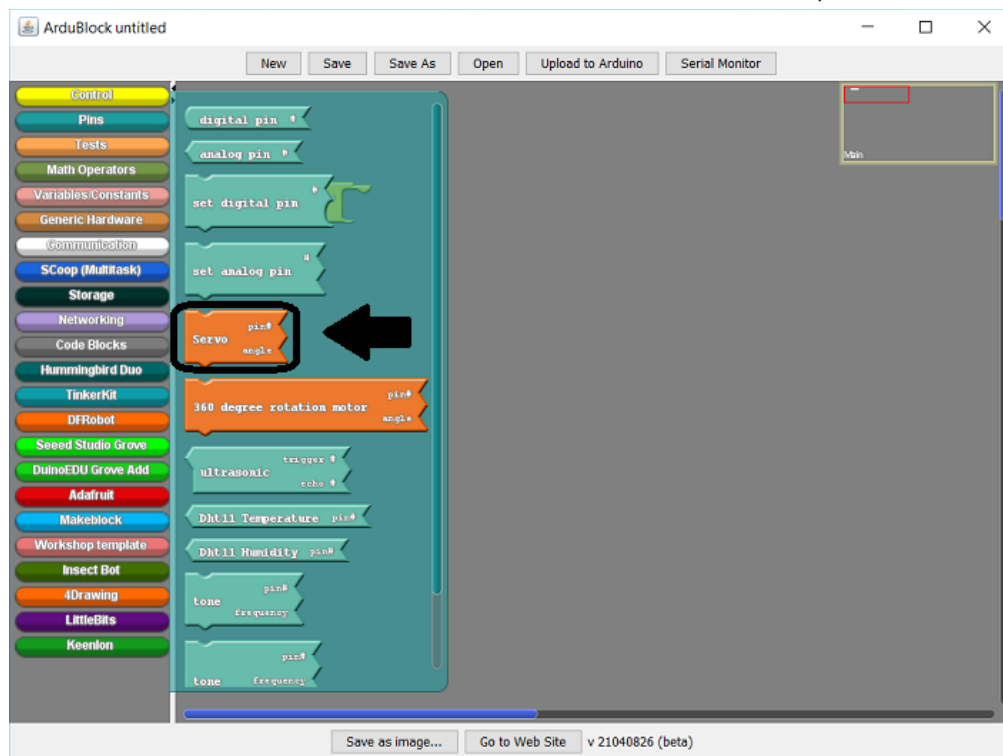
In order to attach our motors to the breadboard we will need to augment the servo motor wires. You will need three jumper wires, preferably yellow, red and black, to correspond with the orange, red and brown wires of the servo motor. Place one end of each of the jumper wires into the ends of the motor wires, being sure to color code the wires appropriately; yellow->orange, red->red, black->brown.

Just like it was recommended in past lessons, try having your students collectively build a breadboard diagram of the circuit based on the schematic you have drawn. Below is a diagram with all of the components we currently have attached to the robot:



Step 3: Getting Our Robot To Wave (45 min)

To control the servo motor we will need the servo block, located in the pins tab:



Upon closer inspection you will see, like some of the other blocks we have used to this point, the servo block expects two input values:



The first, unsurprisingly, refers to the pin the motor has been placed on. In our case this is pin 9. The second is called the angle. The angle refers to the position that the motor will move to, *not the total amount of motion that will take place*. For example inputting an angle of 90 *will not* make the motor move 90 degrees, it will move to whatever position is associated with 90 degrees, no matter how near or far that is from the motor's current position.

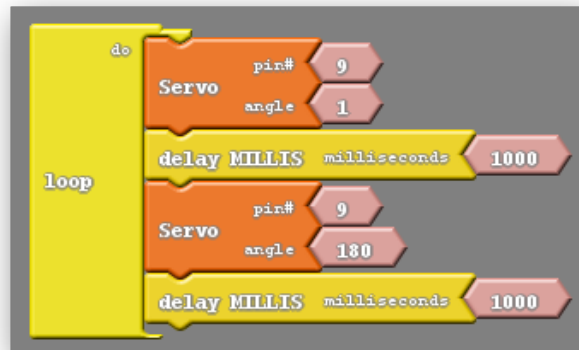
While programming the servo motor, don't forget to consider its physical limitations. The servo motor is capable of rotating in half a circle, 180 degrees. Because of this the angle from 0 to 180 are valid inputs for the servo block. You are welcome to try other angles, it will not damage the motor to do so but you should not expect the motor to behave itself if you do.

With all that being said I think it is time to experiment. Just have your students pick an angle they would like to try and upload their code.

What happens? Many of your students will claim they say movement, however they can not get the movement to repeat, even after uploading their code again. After giving them some time to experiment and note this behavior recommend to the students that they change the angle before uploading again, and to be more precise they should pick an angle that is drastically different than their current angle. At least 30 degrees different. They should notice that the motor moves to a new position, and once again stays there.

This is a good time to reiterate that the angle chosen does not move the motor by that much, but rather moves the motor to the position associated with that angle.

Can we do better than this? Can we do better than repeatedly uploading code to the robot in order to get constant motion from the robot? If fact we can, and doing so won't require knowledge of any new blocks. Much like the set digital pin and tone blocks we can use delay in tandem with the servo block to create repeated effects, such as what is shown below:



The code above will move one of the robot's motors back and forth over the span of two seconds.

Vocabulary

- **Servo Block:** The servo block is the block used to control the servo motors on the robot. It requires two things, the pin the motor is located on and the angle the motor is meant to move to. The angle can accept any number but the physical limitations of the servo motor limit the effective range from 0 to 180.



REFLECTION

Comprehension

1. If the motor has previously been set to 20 degrees and is then set to 90, how many degrees does the motor move?
2. Why are there three wires attached to the servo motor and what do each of them do?

Challenges

What was hard?

Enjoyment

What was fun?

Mindset

What did you learn about yourself? What do you need to work on?

Community

How can what you learned impact those around you?



WEEK 14 OF 16: SERVO MOTORS CONTINUED

Suggested Time: 60-75 minutes

Lesson Overview

Disciplinary Core Ideas

Next Generation Science Standards

- ETS1.B: Developing Possible Solutions: Tests are often designed to identify failure points or difficulties, which suggest the elements of the design that need to be improved. (3-5-ETS1-3)

Learning Target(s)

Technical Skills

- Understanding physical limitations; range of motion (angles), power consumption.

Life Skills

- Perseverance

Essential Questions

1. Can the motors be programmed to move simultaneously? If so, how?

Key Vocabulary

1. None

Additional Resources

Videos

- [Wiring three servo motors](#)
- [Programming multiple servo motors](#)
- [Programming the motors, LED, and buzzer](#)

Documents

- None

Other

- None

Depth of Knowledge Levels Addressed

Level 1: Recall and Reproduction

Level 2: Skills and Concepts

Level 3: Strategic Thinking and Reasoning

Barriers To Learning

- Poor understanding of range of motion, angle position vs. angle of motion

Level 4: Extended Thinking

- Inability to distinguish between known blocks in Ardublock
- Poor understanding of previous coding exercises. In particular the last lesson.

Anticipatory Set

- Students will build upon the learnings of last lesson by making multiple motors move within a single program.

Independent Practice

- Wiring the remaining two motors
- Experimenting with moving multiple motors

Final Assessment, Project, or Product

- Getting the robot move all motors, both one at a time and simultaneously

Lesson Materials (per student)

- ☐ 6 x Jumper wires (preferably yellow, red, black)
- ☐ 1 x Engineering journal
- ☐ 1 x Pencil
- ☐ 1 x Computer with Arduino/Ardublock software
- ☐ 1 x Assembled robot

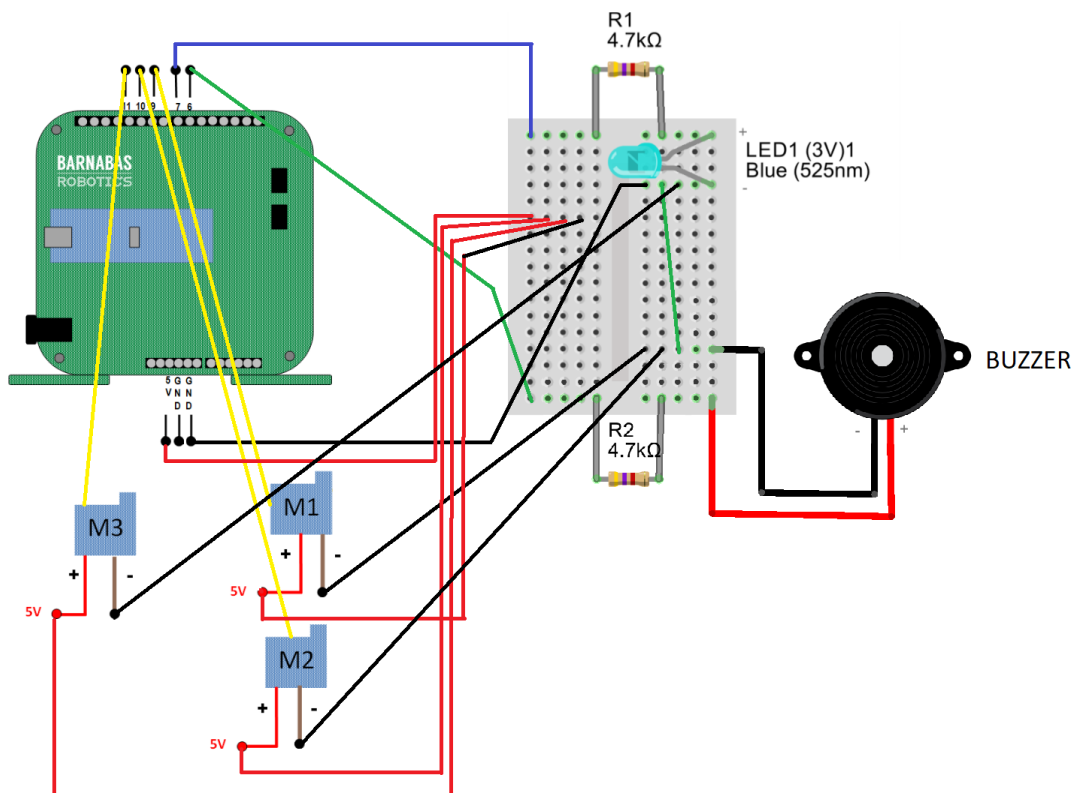


WEEK 14 OF 16: SERVO MOTORS CONTINUED

Lesson Plan

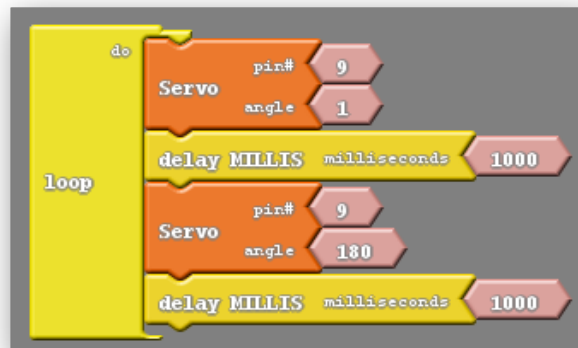
Step 1: Wiring The Other Two Motors (10 min)

The two remaining servo motors will be wired in the same way the first motor was. For each motor grab a yellow, red and black jumper wire and place them in the ends of the motor wires, color coding them like before. Next the red wires (5V) should be placed on the same row of the breadboard as the red (5V) wire of our first motor, and the black wire can be placed on any row of the breadboard that is already home to a black wire. Finally the last two yellow wires must be plugged into pins 10 and 11, like shown below:



Step 2: Experimenting With The Motors (20 min)

With all of the motors wired to the robot getting everything moving is just a couple obstacles away. First I would recommend that the students experiment with the other two motors to find which on pin 10 and which lies on pin 11 as well as find the range of motion for each of the motors. To do this I want to bring back the code we ended on last week:



The students should start here and simply change the pin numbers to 10, then upload and note the behavior of their robot. Afterwards they can do the same but using pin 11 instead. It is important that *both* servo blocks are set on the same pin. Through this process the students will be able to map each pin, 9, 10 and 11, to one of the robots extremities.



Many of the students will find that the range of motion for their motors is not what they desire. For example one of the arms may be waving backwards. The students can correct this by moving the arm to a desired position, *However this can only be done while the robot is unpowered.*

Attempting this while the motor is powered can break the motor.

Step 3: Moving Multiple Motors (30 min)

With all of the motors attached and tested it is now possible to get them all moving. We will start by creating a program that moves two motors independently. The typical error when attempting this is forgetting that each motor should have multiple commands if the goal is getting motion to repeat. A student that makes this error generally only makes a single change from the previous code, resulting in this:



In the code above you can see that both pins 9 and 10 are given commands, however each is only given one command. This will cause each motor to act only once, then rest at whatever position it was told to move to. What the students should do is add servo blocks so that both pin 9 and pin 10 have two servo blocks associated with them.

At this point there is another common pitfall for students. Many students will make the mistake of grouping the same angles with the same pin number, like shown below:



Here you can see that each block associated with pin 9 has been given the angle 1 and each block associated with pin 10 has been given the angle 180. This program behaves identically to the previous one. To get each motor moving repeatedly each motor must be given two distinctly different commands:



Of course we can extend this principle to three motors by adding pin 11 to the mix, giving it two servo blocks of its own and making sure that the angle given is different for each of them.

In addition we can create code that will move all of the motors simultaneously. The key to doing this is to remember how the LED behaved when we failed to use delay blocks. We were unable to see it blink as the two events, turning off and turning on, were so close together. We can intentionally take delay blocks out of the code and rearrange some of the servo blocks to make the action of the motors moving happen immediately after one another. So close together appear to happen at the same time. Below is an example of code that does this:



There are a few things to point out in the code above. First, I have not grouped servo blocks that are using the same pin number together. Each of the two groups of servo blocks have no repeated pin numbers. The second thing we have gone over already, each pair of blocks with the same pin numbers have different angles.

Following those two rules you can make the robot move an many different ways, including moving two motors at a time, rather than all three. Perhaps you combine these things, making the robot wave with a single hand before dancing with all of the motors.



REFLECTION

Comprehension

1. Explain the process of making multiple motors move in the same code. What are some common mistakes that must be avoided?

Challenges

What was hard?

Enjoyment

What was fun?

Mindset

What did you learn about yourself? What do you need to work on?

Community

How can what you learned impact those around you?



WEEK 15 OF 16: ADVANCED CODING

Suggested Time: 60-75 minutes

Lesson Overview

Disciplinary Core Ideas

Next Generation Science Standards

- ETS1.A: Defining and Delimiting Engineering Problems: Possible solutions to a problem are limited by available materials and resources (constraints). The success of a designed solution is determined by considering the desired features of a solution (criteria). Different proposals for solutions can be compared on the basis of how well each one meets the specified criteria for success or how well each takes the constraints into account. (3-5-ETS1-1)
- ETS1.B: Developing Possible Solutions: [At whatever stage, communicating with peers about proposed solutions is an important part of the design process, and shared ideas can lead to improved designs. \(3-5-ETS1-2\)](#)
- Tests are often designed to identify failure points or difficulties, which suggest the elements of the design that need to be improved. (3-5-ETS1-3)
- ETS1.C: Optimizing the Design Solution: Different solutions need to be tested in order to determine which of them best solves the problem, given the criteria and the constraints. (3-5-ETS1-3)

Learning Target(s)

Technical Skills

- Learning to integrate new ideas, subroutines, with old ones, motors, LEDs and buzzers.

Life Skills

- Perseverance
- Teamwork

Essential Questions

1. What advantages does coding with subroutines provide?

Key Vocabulary

1. Subroutine

Additional Resources

Videos

Documents

Other

Depth of Knowledge Levels Addressed

Level 1: Recall and Reproduction

Level 2: Skills and Concepts

Level 3: Strategic Thinking and Reasoning

Level 4: Extended Thinking

Barriers To Learning

- Poor understanding of previous programming concepts
- Inability to grasp abstract concepts

Anticipatory Set

- Students will learn about subroutines as advanced methods of implementing previous coding ideas.

Independent Practice

- Creating several subroutines for individual tasks that have already been covered i.e. waving, blinking, siren noises etc.

Final Assessment, Project, or Product

- Having the robot do several different actions one after the other with the use of subroutines.

Lesson Materials (per student)

- ☐ 1 x Engineering journal
- ☐ 1 x Pencil
- ☐ 1 x Computer with Arduino/Ardublock software
- ☐ 1 x Assembled robot

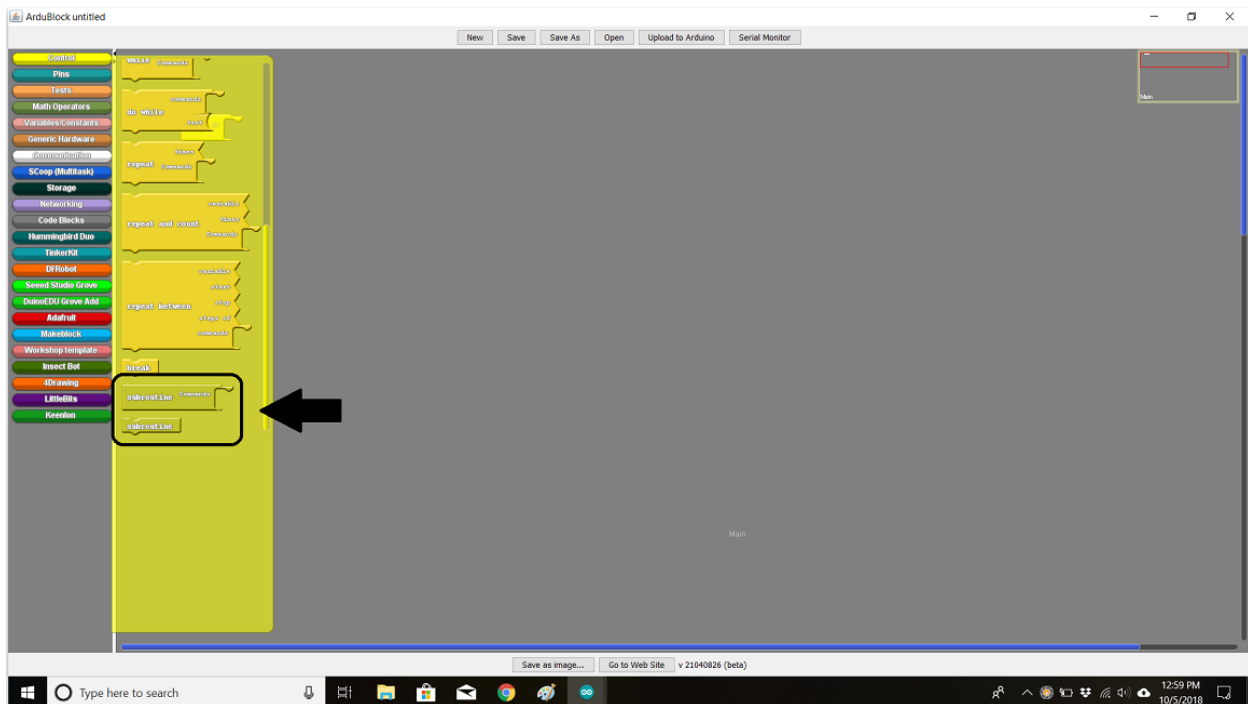


WEEK 15 OF 16: ADVANCED CODING

Lesson Plan

Step 1: Subroutines (60 min)

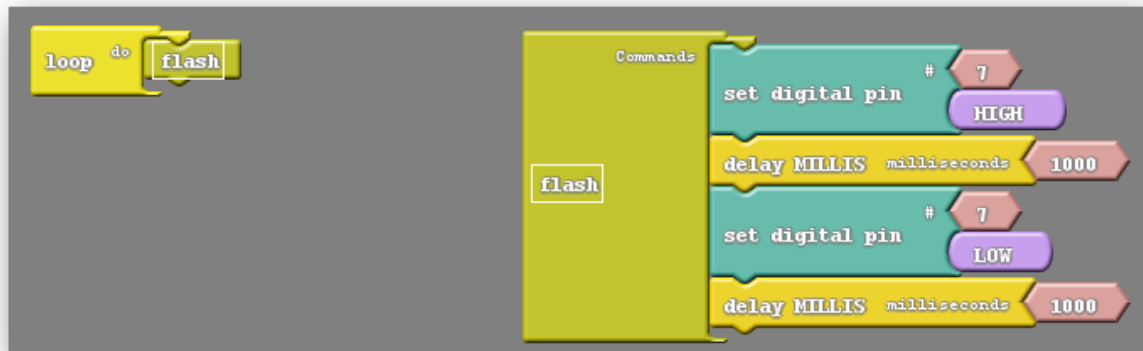
In Ardublock we can code using what are called subroutines, which can be found at the bottom of the control tab as shown below:



The two subroutine blocks available are the subroutine commands block and the subroutine block (which I will refer to as the subroutine run block). Upon closer inspection of the two blocks I can see that only one of them fits into the loop do:



The subroutine commands block acts as a place to hold code. If we subroutine commands block contains other blocks in it we can use the subroutine run block to reference that code later. Here is an example:



Inside the subroutine commands block you can see I have recreated the blinking light code. Also, I have renamed the subroutine commands block to flash. The exact same has been done to the subroutine run block, which also reads flash. When the code attempts to run the loop do sees flash and tries to find a subroutine command block with the same name. If it is successful in finding an identically named subroutine command block it will essentially insert all blocks within that subroutines command block into the code. Looking above you can think of the block inside the loop do as a placeholder for all of the blocks inside the subroutines command block.

The real question is why is this useful? Haven't we just made more work for ourselves? Well, yes and no. In the example above the use of subroutines is unnecessary, but as we aspire to create more and more complex code subroutines become a powerful tool. Take a look at the list of tasks I want my robot to do below:

- 1) Wave its right hand
- 2) Blink
- 3) Make a siren noise for several seconds
- 4) Wave its right hand again
- 5) Dance for a few seconds

Doing this without subroutines will take much work and be difficult to keep organized. However if I take the time to create subroutines for each of these actions I can clean up the code inside the loop do tremendously:



Notice that in the code above some subroutines are used multiple times. This is done while only making one subroutines command block per each distinctly named subroutine. You can see that in this case, where some subroutines are run multiple times, having subroutines is much better than not. In addition to that the code is much easier to read and understand.

Vocabulary

- **Subroutine:** A set of instructions designed to perform a frequently used task. In Ardublock the subroutine blocks consist of the subroutine commands block, which holds the instructions, and the subroutine (run) block, which fits into the loop do.



You may have noticed that I avoided calling the LED based subroutine blink, calling it flash instead. This is because blink is a named command in Arduino already, and naming a subroutine blink causes a conflict when the Ardublock code is translated to Arduino. You could however name the subroutine Blink, as subroutines are case sensitive.



The process of creating and calling subroutines is common amongst every programming language. Typically in typed programming languages subroutines are given the name functions instead. For the students planning to continue learning about programming beyond this class functions will become a constant.



REFLECTION

Comprehension

1. Using subroutines is a two step process. Explain the two steps.
2. How is naming subroutines significant?

Challenges

What was hard?

Enjoyment

What was fun?

Mindset

What did you learn about yourself? What do you need to work on?

Community

How can what you learned impact those around you?



WEEK 16 OF 16: PUTTING IT ALL TOGETHER

Suggested Time: 60-75 minutes

Lesson Overview

Disciplinary Core Ideas

Next Generation Science Standards

- ETS1.A: Defining and Delimiting Engineering Problems: Possible solutions to a problem are limited by available materials and resources (constraints). The success of a designed solution is determined by considering the desired features of a solution (criteria). Different proposals for solutions can be compared on the basis of how well each one meets the specified criteria for success or how well each takes the constraints into account. (3-5-ETS1-1)
- ETS1.B: Developing Possible Solutions: [At whatever stage, communicating with peers about proposed solutions is an important part of the design process, and shared ideas can lead to improved designs. \(3-5-ETS1-2\)](#)
- Tests are often designed to identify failure points or difficulties, which suggest the elements of the design that need to be improved. (3-5-ETS1-3)
- ETS1.C: Optimizing the Design Solution: Different solutions need to be tested in order to determine which of them best solves the problem, given the criteria and the constraints. (3-5-ETS1-3)

Learning Target(s)

Technical Skills

- Integration of multiple technical skills

Life Skills

- Perseverance
- Teamwork

Essential Questions

Key Vocabulary

Additional Resources

Videos

- [Getting your robot to sing, light up and dance!](#)

Documents

Other

Depth of Knowledge Levels Addressed

Level 1: Recall and Reproduction

Level 2: Skills and Concepts

Level 3: Strategic Thinking and Reasoning

Level 4: Extended Thinking

Barriers To Learning

- Poor understanding of previous programming concepts

Anticipatory Set

- None

Independent Practice

- Draft a plan for the final code for the robot.

Final Assessment, Project, or Product

- Implement drafted code and troubleshoot appropriately. Sharing their robot with the class.

Lesson Materials (per student)

- ☐ 1 x Engineering journal
- ☐ 1 x Pencil
- ☐ 1 x Computer with Arduino/Ardublock software
- ☐ 1 x Assembled robot



WEEK 16 OF 16: PUTTING IT ALL TOGETHER

Lesson Plan

Step 1: Putting It All Together (45 min)

In this lesson there will be no new concepts introduced. Instead this lesson is meant to be treated as more of an *open lab* style class, meaning the students work independently for the majority of class. What I would recommend is to have each of the students draft an ordered list of things they want their robot to do, much like the list shown in the last lesson, which I will show again here:

- 1) Wave its right hand
- 2) Blink
- 3) Make a siren noise for several seconds
- 4) Wave its right hand again
- 5) Dance for a few seconds

It is likely that many of your students will have all the subroutines they place to use premade from the last lesson. They should still be encouraged to create a few new subroutines to add to their repertoire. Their goal should be to have their code made completely using subroutines and delays.

Step 2: Share (15 min)

With the students more or less done with their creations, It is time for the class to present. Ultimately how you have your students present is up to you, and how you choose to handle presentations may change the overall shape of this lesson, but I will be giving my suggestion for student presentations. Rather than have the students present one by one in front of the class I prefer to create a science fair type atmosphere. Have each of the students set up their robots and computers side by side with their code visible. Encourage the students to visit other students work spaces and engage each other. The goal is to create an organic sharing of ideas where students are intrigued enough by others work that they want to implement some of the things they have seen.



REFLECTION

Comprehension

1. What is your favorite aspect of your own robot?
2. What is your favorite aspect of someone else's robot?

Challenges

What was hard?

Enjoyment

What was fun?

Mindset

What did you learn about yourself? What do you need to work on?

Community

How can what you learned impact those around you?

Appendix A: References

- https://static.pdesas.org/content/documents/M1-Slide_19_DOK_Wheel_Slide.pdf

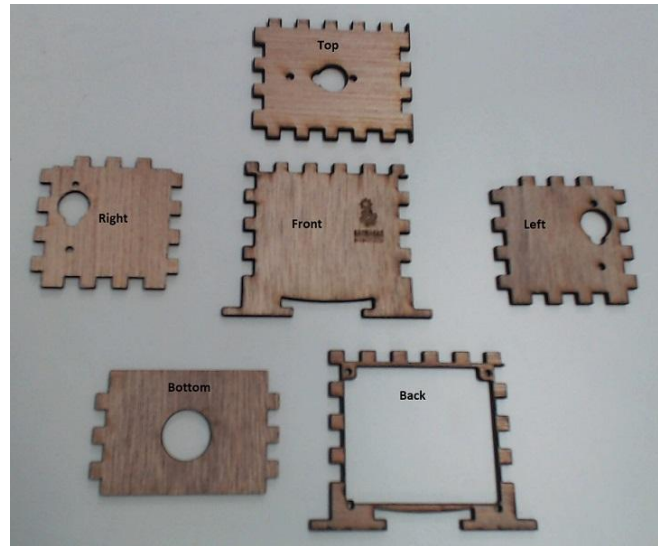
Appendix B: Additional Resources

Appendix C: Wood Bot Addendum

Step 1: Identifying The Parts

The laser cut wood version of the barnabas bot has a body made up of distinct parts that fit together like a jigsaw puzzle. Because of this it is very important that we can easily identify all of the parts as well as how they should be oriented.

- Array all of the wood pieces and make sure that you have all of the pieces shown in the picture to the left.



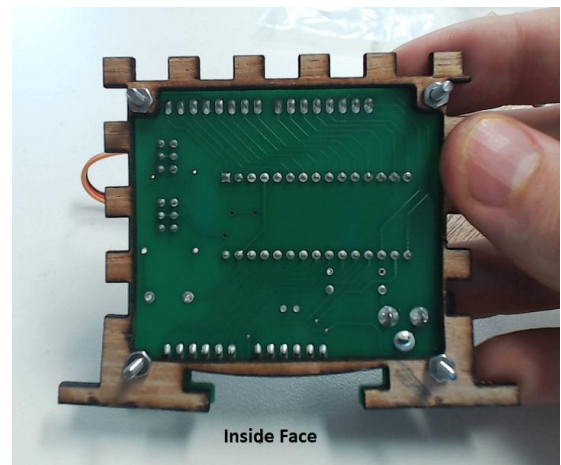
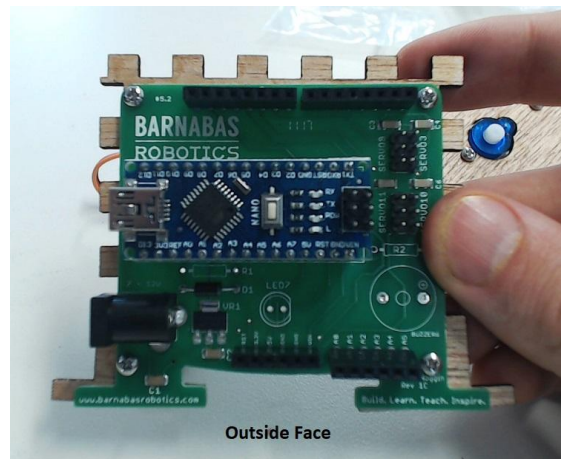
- Without any glue, put the wood pieces together. It should become fairly obvious that the full robot body will only fit together if the parts are oriented a certain way. This means that for each piece there is an inside edge and an outside edge. After fitting all the pieces together mark the inside faces with a pencil.



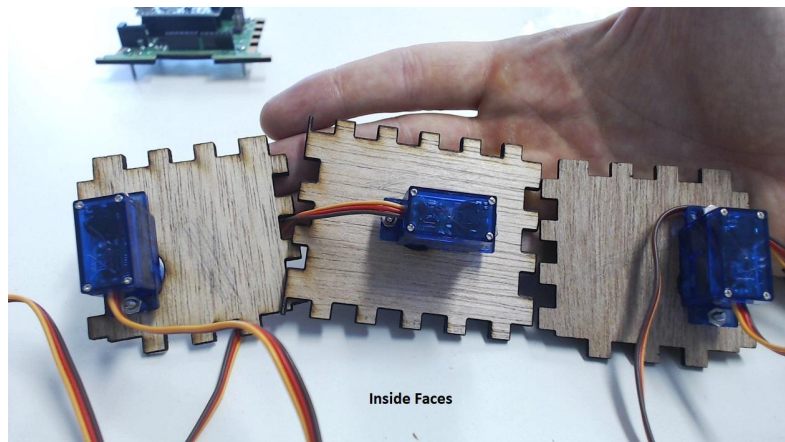
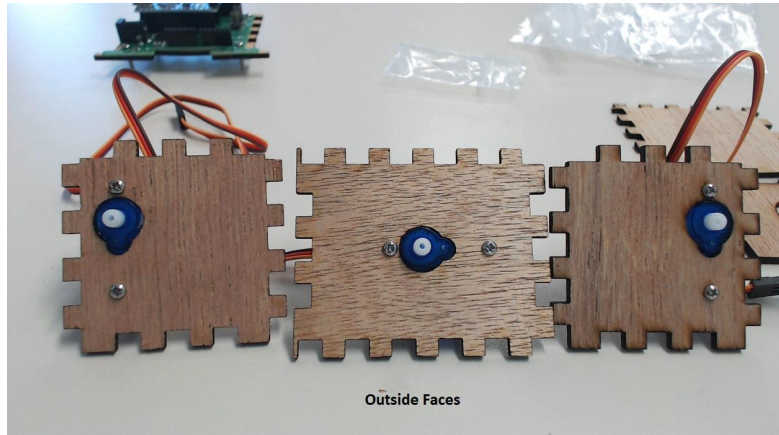


Step 2: Placing The Noggin and Servo motors

- Now, disassemble the body and fasten the noggin to the back plate using the screws provided with the noggin on the outward face.

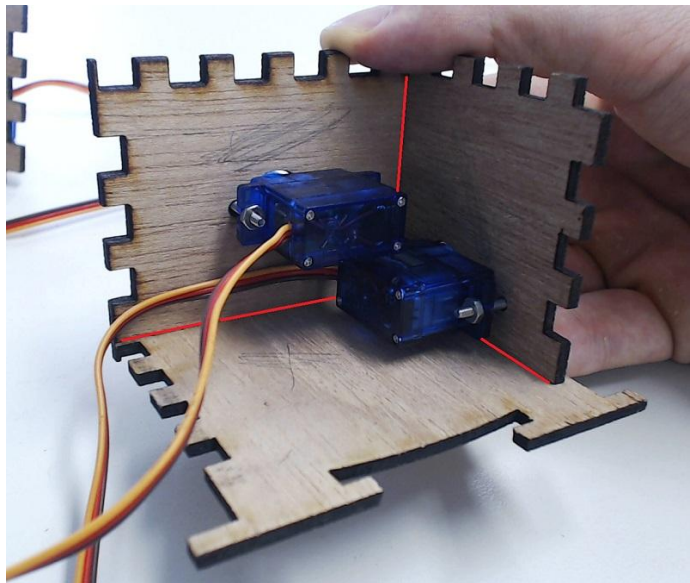


- Likewise fasten the motors to the left, right, and top plate with the motor placed on the inside face of each plate.



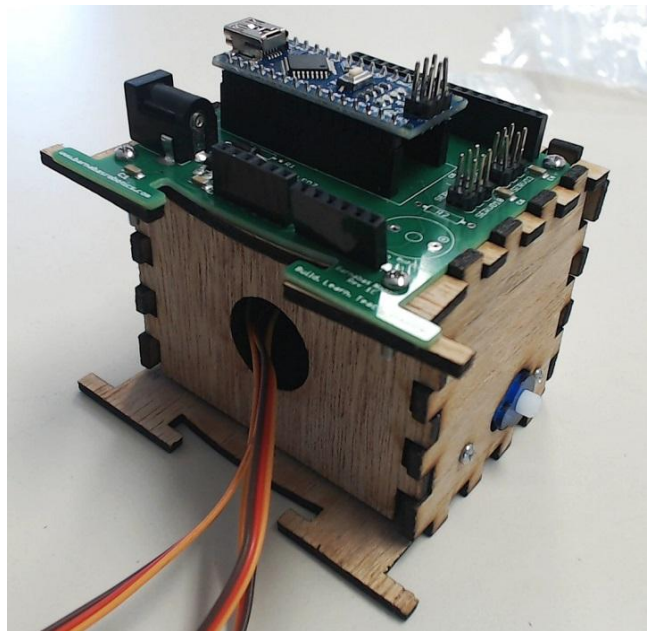
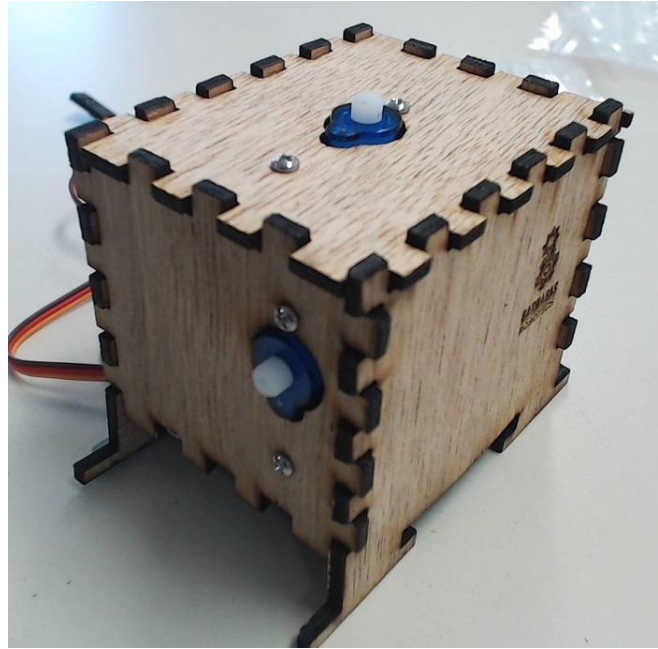
Step 3: Gluing The Robot Together

- Begin to reassemble the robot body, this time applying glue to keep everything in place. There are several ways to glue the robot together and I will only be highlighting one. I generally will apply glue to the inside seam between two plates like shown in the picture to the right. The red lines indicate where the glue is to be placed. This can be done with elmers glue but can also be done with certain super glues such as crazy glue. In either case it may be difficult to reach every corner of the robot and I would



recommend using a skinny popsicle stick or a small brush to apply the glue. If you choose to use super glue be sure to handle the glue yourself and keep out of the reach of your students. Be sure to add the back plate after adding the bottom plate.

- Your robot assembled up to this point should look like the pictures to the right.



Step 4: Decorating And Adding The 3D Printed Parts

- This would be a great point to pivot to decoration. The students can use colored pencils, markers or sharpies to color both the wood parts as well as the 3D printed parts

we have yet to add to our robot. Of course if you are using slow drying glue this step should wait.

- Fasten the 3D printed head and arms to the motors. This is conducted in exactly the same way as the week 11 lesson, so more in depth instructions are provided there.

