Black Horse Pike Regional School District
580 Erial Road, Blackwood, NJ 08012

Robotics

COURSE OF STUDY

Technology Education Department

Written By:
Dustin Keyser

Date:
Summer 2022

Supervisor:
Steve Arena

Approved by:
Marcie Geyer, Director of Curriculum & Instruction

ROBOTICS ENGINEERING

HIGH SCHOOL NAME

2020-2021 Course Syllabus

TEACHER

E-MAIL

PHONE #

REMIND CODE

CLASSROOM CODE

COMMON TIME AVAILABILITY

_ and _ days in _-___.

I am not available any other day, but can meet after school, if you schedule it with me in advance.

COURSE DESCRIPTION

11-12 Graders - MUST HAVE PASSED ALGEBRA 2 AND ONE TECH ED CLASS

5 Credits

In Robotics Engineering, you will design, engineer, program, and build robots to compete against others in the class in fun, hands-on design activities. As the year progresses and projects get more involved, you will advance from manual control to programmable control, from pre-fabricated parts to custom 3D printed parts, and from pre-soldered circuits to creating simple circuits of your own.

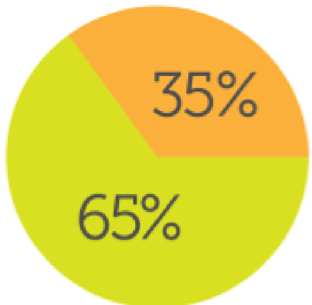
UNITS COVERED

- Electrical Safety
- Electrical Components, Simple Circuits, and Basic Physics
- Circuit Construction and Analysis Using Tools Like TinkerCAD
- Coding Basics (Scratch) Using Sphero Bots
- 3D Printing and Rapid Prototyping
- RC Control Using Tetrix Robots
- Programmable Control with Tetrix Ardublocky (C++)

MATERIALS NEEDED

- Pencil (bring everyday)
- Two-Pocket Folder (keep in the classroom)
- You may be required to get batteries for certain kits/electrical circuits

GRADING



CLASSWORK (65%): Anything you make or produce falls under this category including written assignments, quizzes, sketches and brainstorming activities, any computer-based work, and anything you build from your prototypes to your final designs.

PARTICIPATION (35%): You are expected to actively participate each and every day. Over a third of your grade is participation! Below are some pointers to help you succeed in class and earn full participation points every week:

LATE WORK

- Late work is -10% each day (not each class, but each day) it is late, up to -50% (5 days late). Work can still be turned in after that, but will be -50%.
- Up to teacher discretion.

ABSENCES

- If you are absent, it is your responsibility to e-mail me and check Google Classroom.
- You will get extensions on assignments equal to the number of days you were out
- If you know you're going to be out, notify me ahead of time so I can help you with the classwork

TOP 10 WAYS TO EARN FULL PARTICIPATION POINTS

1. Keep phone and headphones away for the entire class period
2. Keep quiet and pay attention during lectures, lessons and demonstrations
3. Ask questions if you're not sure what to do. If I am busy with someone else, try and look up the answer on your own or ask a classmate or partner (but don't do nothing)
4. Each class we will have daily/weekly checkpoints. Make sure you know what they are and work to meet those checkpoints
5. When prompted to get to work, you should get to work within just a couple minutes. Any longer and you will lose participation points
6. Be productive and try your best. You should be working on your projects for this class for the majority of the period. Breaks are ok, but should be short and limited
7. Use school appropriate language and be mindful of your classmates
8. Arrive to class on time, prepared with all necessary materials and sit in your seat
9. Use tools, machinery, and classroom equipment correctly and safely
10. Have fun!

DESIGN & TECHNOLOGY DEPARTMENT

PERFORMANCE CHART

	ADVANCED	PROFICIENT	BELOW AVERAGE
GRADING GUIDE	A	B-C	D-F
EFFORT AND USE OF CLASS TIME **(Group or Individual)**	Extra effort during and after class time is put into project. Student(s) modeled exceptional behavior, were always on task, followed all safety rules, and helped others.	Consistent effort is put into the project during class time. Student(s) modeled good behavior, but was not always on task and misused equipment.	Inconsistent effort during class time. Student(s) did not use class time wisely, misused tools and machinery, and, as a result, missed checkpoints, deadlines and due dates.
ACCURACY AND NEATNESS	Project is prepared neatly and carefully. All measurements are accurate. Project is aesthetically pleasing and well built.	Project is fairly neat. Measuring is mostly accurate. Project is good. There is room for improvement.	Project is prepared with little care and lacks neatness. Project looks rushed and doesn't work as intended. Lots of room for improvement.
CREATIVITY	Project is original and imaginative. Design is unique, innovative and well thought out.	Project has some original elements. Design is somewhat clever but not entirely unique.	Project lacks creativity and thought. Design is not original and is more or less a copy of an existing one.
FOLLOWING INSTRUCTIONS, SPECIFICATIONS AND CONSTRAINTS	All project instructions have been followed. Every requirement has been met and exceeded.	Some project instructions and requirements met, but not all.	The majority of project instructions and requirements were not followed, have not been met, and project is incomplete.
DEMONSTRATES UNDERSTANDING	Student is extremely knowledgeable of project concepts and is able to help others.	Student displays knowledge of most concepts, methods and/or practices involved in the project.	Student lacks knowledge about project concepts, methods and practices.

Robotics Engineering

2022-2023 School Calendar

1. Basic Electronics and Components
2. Basic Block and Text Coding
3. Advanced Electronics, Components, and Circuitry Prototyping
4. Soldering
5. Additive Manufacturing
6. Technological Design Process (District Tech Challenge)
7. Advanced Text Coding for Robotic Control
8. Robotics Design

September '22

Su	M	Tu	W	Th	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

October '22

Su	M	Tu	W	Th	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

November '22

Su	M	Tu	W	Th	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

December '22

Su	M	Tu	W	Th	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

January '23

Su	M	Tu	W	Th	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

February '23

Su	M	Tu	W	Th	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

March '23

Su	M	Tu	W	Th	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

April '23

Su	M	Tu	W	Th	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

May '23

Su	M	Tu	W	Th	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

June '23

Su	M	Tu	W	Th	F	S
				1	2	3
4	~~5~~	~~6~~	~~7~~	~~8~~	~~9~~	10
11	~~12~~	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Unit Summaries

Robotics

1. **Basic Electronics and Components** - Students will start off the year learning about the basics of electricity at the atomic level and look at some of the most common electrical components, including batteries and cells, wires and cables, lamps and LEDs, switches, resistors, and fuses and circuit breakers. Students will create a variety of series and parallel circuits using many of these components and will take voltage, current and amperage readings using a multimeter and use Ohm's Law to calculate the relationship between voltage, current and resistance.

2. **Basic Block and Text Coding** - In this unit, students will be introduced to Block Coding. Block coding turns programming into a drag-and-drop process by converting text based code into visual blocks. Each block contains real code and when they're combined together, students can experiment, explore, and create animations and games. Students will use Block Coding to control Sphero Bolt robots through a series of mini challenges. After a few of these hands-on activities, students will delve deeper into Block Coding with VEXcode VR: a web-based application where students can code and control a robot in a virtual setting. Students will start by making the robots travel forwards and backwards, but as they progress through the modules, students will code and control servos, line-finding sensors, and Infrared (IR) Sensors. And then, with the basics of coding a robot down, students will be ready for their first taste of text-based coding. They will continue to use VEXcode VR, but instead of blocks, students will use a programming language called Python to learn Computer Science concepts like project flow, loops, conditionals, and algorithms.

3. **Advanced Electronics, Components, and Circuitry Prototyping** - Now that students have some knowledge of basic electronics, circuitry, and some text-based coding under their belt, it is time to step things up a bit with the Arduino. The Arduino is a microcontroller, which means that it is a tool that is able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. Students will use the Arduino to learn yet another programming language, C++, and will create some really fun and unique projects, like a motorized pinwheel and a keyboard instrument. Students will use TinkerCAD to model, test, and refine their circuits before building the actual prototype, just like Electrical Engineers do on a daily basis.

4. **Soldering** - In this unit, students will learn how to safely solder, with particular emphasis on through-hole components on a circuit board. Students will learn to read schematic diagrams, identify components, and place these components accordingly on their Printed Circuit Board (PCP).

5. **Additive Manufacturing** - Once students have successfully soldered their circuits, it's time to create a custom housing assembly for their circuits to protect it from damage. Students will take measurements using a micrometer, learn about fastener sizing, pitch, and threads, and will create a multi-part assembly to encase their projects. Once printed and assembled, students will share their designs with the class.

6. **Engineering Design Process (District Tech Challenge)** - In this unit, students will work collaboratively in small teams to solve an open-ended design challenge using the steps of the Engineering Design Process (EDP). In lower level Design & Tech classes, students are walked through each step of the process at a slower and more in-depth pace. But in higher level classes, like Robotics, students are free to follow the steps of the TDP at their own pace and must document their progression through the TDP by means of a Design Journal. Once students have created a viable solution, they will refine, retest, and rebuild their design in hopes of being crowned the district champion as they compete against other teams from the Technology Education classes in the district.

7. **Advanced Text Coding for Robotic Control** - Students will build and program a Tetrix robot to perform autonomous tasks using C++ within the Arduino IDE software. These tasks will start out fairly simple - Drive forward. Stop. Turn right. But as students progress through, they will add more hardware and sensors to their robots and the tasks will get more and more complex - Use the Ultrasonic Sensor to sense obstacles to avoid them, use the Line Finder

sensor to get your robot to parallel park, or combine the two to have a fully autonomous robot successfully navigate itself through a maze.

8. **Robotics Design** - In this unit, students will work in teams to synthesize everything they learned throughout the year and will be given open-ended design challenges where they will compete against others in the class. Some challenges will require additional sensors and circuitry, some might require a custom fabricated part, some might require Remote Control (RC) and Bluetooth capabilities. The possibilities are endless!

Programs: Arduino IDE, Sphero Edu, VexCODE VR, Robotify (Tetrix), TinkerCAD, Autodesk AutoCAD, Onshape.

Course Expectations

1. Come to class each day willing to participate, prepared to learn, and ready to succeed.
2. Phones and other electronic devices should be kept away the entire time.
3. Use your resources when you are stuck (class notes or handouts, posts on Google Classroom, ask a classmate, search for the solution online, etc.)
4. When prompted to get to work, do not wait. Get to work right away!
5. Engineering is a process. Project will often take weeks or months to complete. Understand that you might not have the answers today, but if you keep at it and persist through, you will succeed.
6. Strive for accuracy and focus on craftsmanship.
7. See each project through to the end. Incomplete work will not be accepted.
8. Share ideas, help build, and equally contribute to group-based projects.
9. Practice proper attitude and safe discipline at all times.
10. Ask questions and have fun!

Course Skills

1. Use tools and machines confidently, correctly and safely.
2. Identify common electronic components and use them to prototype circuits using a breadboard and an Arduino Uno microcontroller.
3. Test, analyze, troubleshoot and fix circuits using computer simulations and multimeters.
4. Solder thru-hole components on a printed circuit board (PCB).
5. Code virtual and physical robots using a number of computer programming languages.
6. Test, analyze, troubleshoot and fix robots using C-based computer programming languages.
7. 3D model parts, assemblies, sub-assemblies, and hardware and document these designs on a series of Engineering Drawings, using industry-standard practices.
8. 3D print, laser engrave, and use other manufacturing techniques to fabricate original parts for robotic designs.
9. Apply the steps of the Engineering Design Process (EDP) to every design challenge.
10. Collaborate with others to apply and analyze STEAM-related concepts to solve complex open-ended robotic design challenges.

Resources:

Tetrix PRIZM Programming Guide. Content advising by Paul Uttley, Pamela Scifers, Tim Lankford, Nevin Jones, and Bill Holden. 2018.

Technology Education: Learning by Design. Second Edition. Michael Hacker, David Burghardt

Unit 1:	Basic Electronics and Components

Unit Summary
Students will start off the year learning about the basics of electricity at the atomic level and look at some of the most common electrical components, including batteries and cells, wires and cables, lamps and LEDs, switches, resistors, and fuses and circuit breakers. Students will create a variety of series and parallel circuits using many of these components and will take voltage, current and amperage readings using a multimeter and use Ohm’s Law to calculate the relationship between voltage, current and resistance.

Essential Questions	**Enduring Understandings**
• What are batteries and cells? • What are diodes? • How do LEDs work? • How do switches work? • How are resistors classified?	• Each component in a circuit serves a very particular purpose and is designed to perform a certain task. Knowing what these purposes/tasks are is fundamental in understanding electronics and how we harness and use electricity to meet human wants and needs.
• What is electricity? • How does electricity work on the atomic level? • What does it mean when something is positive (+) or negative (-)? • What is the difference between a series and parallel circuit?	• It can be difficult to understand what electricity is and how it works because we cannot always see what’s happening. We might see lights and motors turn on and off or feel heat on components and understand this to be electricity, but understanding how electricity works on an atomic level is fundamental in learning about circuits • Some components only work when hooked up a certain way in a circuit.
• Why is it important to make a schematic before building a physical circuit? • What is TinkerCAD?	• Electronic components can be costly, but also quite dangerous if hooked up improperly. It is always a great idea to make schematics, build virtual circuits, and double check components before hooking up to power.
• How do components work differently in series vs. parallel circuits? • What is the significance in understanding Ohm’s Law? • What is the formula for Ohm’s Law?	• Some components work differently when hooked up in series vs. parallel. For instance, if one LED in a series circuit goes out, the rest will too. But in parallel, if one burns out, the others will remain on • Ohm's Law is a formula used to calculate the relationship between voltage, current and resistance in electrical circuits.
• How is a multimeter used to test for continuity? • How are voltage, current and resistance measured?	• A multimeter is a tool used to test circuits for continuity, voltage, current and resistance levels, and helps electricians and electrical engineers troubleshoot and fix problems within a circuit.

Behavioral Objectives/Learning Targets	Standards (NJSLS)
• Identify some of the most basic components and provide a summary (in the student's own words) of how each component works.	8.2.2.ITH.1-2 8.2.5.ITH.1
• Describe how electricity works on the atomic level using terms like: *flow, electrons, positive,* and *negative.* • Describe how electricity works in both a series and a parallel circuit.	8.2.5.NT.4
• Build simple series and parallel circuits using TinkerCAD, then model these circuits using physical components.	8.2.5.ED.2
• Build simple circuits and explain how components like LEDs, switches, and resistors behave differently in series vs. parallel. • Build simple circuits and explain how voltage, current and resistance change in series vs. parallel.	8.2.2.NT.1 8.2.5.ED.1
• Correctly use a multimeter to measure voltage, current and resistance. • Troubleshoot and fix circuits that are not working as intended.	8.2.5.NT.1

Interdisciplinary Connections
STEAM, Physics, Materials Science, Algebra, English

21st Century Skills
Critical thinking, communication skills, creativity, problem solving, perseverance, information literacy, technology skills and digital literacy, self-direction, social skills, literacy skills, thinking skills

Writing Assignments
Students will complete a series of notes using fill-in-the blank Google Slides presentations. This will be printed out for students with slides on one side and a notes section on the other. Students will complete these for the six sections of components. Students will also complete notes on series and parallel circuits and will submit a written analysis of each circuit, detailing how components work/behave in a circuit and how voltage, current and resistance affect that circuit. To round out the unit, students will be given a schematic design where they will have to model and build a series and parallel circuit from scratch and provide a written summary of how they hooked that circuit up and answer questions related to the voltage, current and resistance readings they take and analyze from their multimeters.

Activities, Instructional Strategies, and Assignments
Since this is the first unit, it will be somewhat heavy on notes. Videos, animations and pictures will be used as often as possible, but there are a lot of components and a lot of background information students need to master before moving on. 1. Students will start by researching components on their own, then will meet with partners to compare answers before reviewing this material as a whole class. These will serve as the class notes and are broken down as follows: a. Batteries and Cells b. Wires and Cables c. Lamps and LEDs d. Switches e. Resistors f. Fuses and Circuit Breakers 2. Students will take a closer look at resistors and will learn about resistor color codes as well as metric conversion. Students will complete: a. Metric Conversion Assignment b. Missing Values of Resistor Color Codes Assignment c. Calculating Resistor Color Codes Assignment 3. Students will then complete notes on Series vs. Parallel Circuits in a similar fashion to their notes on the electrical components. 4. Students will learn about Ohm's Law and will use algebra to determine voltage, current and resistance values in series and in parallel circuits. 5. Students will then begin a collection of Series vs. Parallel lab assignments. These labs start easy, with a power source and single load, like a bulb, and progressively get more and more challenging. With each lab, students must read schematics, hook their circuits up correctly, and then answer review/analysis questions. 6. For the final part of this unit, students will be given a schematic of a series and a parallel circuit. Students will first make these circuits on TinkerCAD, take voltage, current and resistance readings, and submit that for approval. Once approved, students will build the circuit using physical components, take readings with the multimeter, and submit a reflection/analysis of that circuit.

Accommodations and Modifications (BHPRSD Accommodations and Modifications)
• Provide study guides and support outside of class time to review before assessments (common time or after school) • Build background knowledge of content and vocabulary from familiar contexts prior to readings • Use mental models to building understanding through familiar contexts • Provide oral & written instructions • Incorporate multimedia/audio visual representation (YouTube, Discovery Education, TV Show parodies, etc.) to build understanding • Use graphic organizers to guide notes, brainstorming, pre-writing, project planning, and test preparation • Model through processes during assignments and elicit student-generated thoughts to determine gaps in understanding • Highlight, bold, or underline main ideas in readings and in directions for writing assignments in the curricular areas • Provide guiding questions to complete during the activity • Provide chunking of instructional notes and activities to allow for formative assessment (checks for understanding) before moving on to the next stage • Choose cooperative learning groups to ensure effective work, maximize productivity and support socialization • Provide demonstrations, utilize pictures, or graphic to assist visual learners to support written text information

- Include oral discussions, oral presentations, group collaboration, or other oral delivery methods to support auditory learners
- Utilize hands-on activities, movement or rhythmic experiences to engage tactile/kinesthetic learners
- Provide chunking of assignments into manageable steps, including checklists that clarify directions for assignments
- Highlight distinctive features/key concepts
- Review, repeat, and clarify directions
- Chunk sections of assessment
- Allow for partial credit, when appropriate

Formative Assessments

- Electrical Components Research and Notes:
  - Batteries and Cells
  - Wires and Cables
  - Lamps and LEDs
  - Switches
  - Resistors
- Resistor Identification Hands-on Assignment
- Series vs. Parallel Research, Notes, and Lab Assignments
- Multimeter Hands-on Assignment

Summative Assessments

- Final series and parallel circuits from a schematic diagram
- Final written analysis of series and parallel circuits

Performance Assessments

- Safely utilize tools and machines

RETURN TO CALENDAR

RETURN TO UNIT SUMMARIES

Unit 2:	Basic Block and Text Coding

Unit Summary
In this unit, students will be introduced to Block Coding. Block coding turns programming into a drag-and-drop process by converting text based code into visual blocks. Each block contains real code and when they're combined together, students can experiment, explore, and create animations and games. Students will use Block Coding to control Sphero Bolt robots through a series of mini challenges. After a few of these hands-on activities, students will delve deeper into Block Coding with VEXcode VR: a web-based application where students can code and control a robot in a virtual setting. Students will start by making the robots travel forwards and backwards, but as they progress through the modules, students will code and control servos, line-finding sensors, and Infrared (IR) Sensors. And then, with the basics of coding a robot down, students will be ready for their first taste of text-based coding. They will continue to use VEXcode VR, but instead of blocks, students will use a programming language called Python to learn Computer Science concepts like project flow, loops, conditionals, and algorithms.

Essential Questions	**Enduring Understandings**
● How does information travel between inputs and outputs? ● What are some examples of input devices on the robots? ● What are some examples of output devices on the robots?	● Sensors and other hardware may be connected to other devices to form a system as a way to extend their capabilities. ● Software and hardware work together as a system to accomplish tasks (sending, receiving, processing, and storing units of information).
● What are algorithms? ● How is it that different algorithms can be used to solve the same problem?	● Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific use than others, or may be more efficient.
● What are variables? ● What are sequences? ● What are loops? ● What are conditionals?	● Programming languages provide variables, which are used to store and modify data. ● A variety of control structures are used to change the flow of code, like sequences, loops, and conditionals.
● Why might it be beneficial to break a large task down into smaller, sub-tasks?	● Code can be broken down into smaller parts to facilitate design, implementation, and review.
● What if our code is correct, but the robot is not responding as expected? ● How can we find and fix errors within our code?	● Writing code is an iterative process that requires constant implementation, testing and review. ● When troubleshooting robots, we have to look at the written code as well as the inputs, outputs, and even the physical structures that make up the robot we are controlling - and maybe even the controller itself.

Behavioral Objectives/Learning Outcomes	Standards (NJSLS)
• Model how electronic devices (sensors and other hardware) connect to other components to form a system. • Explain how information moves between the software and the hardware.	8.1.5.CS.1-2
• Compare and refine multiple algorithms for the same task and determine which is the most appropriate.	8.1.5.AP.1
• Create and write code that uses clearly named variables to store and modify data. • Create and write code that includes sequences, events, loops, and conditionals.	8.1.5.AP.2-3
• Break down large tasks into smaller, more manageable sub-tasks.	8.1.5.AP.4
• Develop programs using an iterative process, implement the program design, and test the program to ensure it works as intended. • Troubleshoot and fix errors in code, inputs, and/or outputs.	8.1.5.AP.6 8.1.5.CS.3

Interdisciplinary Connections
Computer Science, STEAM, Physics, Algebra, English

21st Century Skills
Critical thinking, communication skills, creativity, problem solving, perseverance, collaboration, information literacy, technology skills and digital literacy, media literacy, self-direction, social skills, literacy skills, innovations skills, thinking skills

Writing Assignments
Students will be doing a lot of coding in the second half of this unit using Python, a text-based coding language. Students will also be required to complete a series of Google Docs and/or forms as entrance and/or exit tickets. Some of these will be fairly specific and require short answers. Some are open-ended and will require a more detailed response. At the end of the unit, students will write a Reflection Essay on what they learned, problems they faced in the unit, how they overcame those problems, and how they can use this experience to help them in future units.

Activities, Instructional Strategies, and Assignments

1. Students will begin this unit with a fun, hands-on mini unit on an Introduction to Block Coding using the Sphero Bolt Robots, which students can program and control right from their Chromebooks. Students will be led through a series of coding challenges, with each new challenge building off of knowledge learned in previous challenges
    a. Students will follow along with lectures, videos, and hands-on demonstrations
    b. Students will start off the first few challenges working alone and the challenges will progressively get more interactive and collaborative
2. Students will use vexCODE VR to code and control robots in a virtual setting. Students will start by making the robots travel forwards and backwards, but as they progress through the modules, students will code and control servos, line-finding sensors, and Infrared (IR) Sensors
    a. Students will follow along with lectures, videos and demonstrations
    b. For some modules, students will submit screenshots of their code, videos of their robot performing specific tasks, or will have to submit a Google Form Quiz or answer a series of questions on a Google Doc and submit to Google Classroom as entrance and/or exit tickets
3. With the basics of coding a robot down, students will be ready for their first taste of text-based coding. They will continue to use VEXcode VR, but instead of blocks, students will use a programming language called "Python" to learn Computer Science concepts like project flow, loops, conditionals, and algorithms
    a. Students will follow along with lectures, videos and demonstrations
    b. For some modules, students will submit screenshots of their code, videos of their robot performing specific tasks, or will have to submit a Google Form Quiz or answer a series of questions on a Google Doc and submit to Google Classroom as entrance and/or exit tickets

Accommodations and Modifications (BHPRSD Accommodations and Modifications)
● Provide a variety of concrete examples from familiar contexts ● Provide study guides and support outside of class time to review before assessments (common time or after school) ● Build background knowledge of content and vocabulary from familiar contexts prior to readings ● Use mental models to building understanding through familiar contexts ● Provide oral & written instructions ● Incorporate multimedia/audio visual representation (YouTube, Discovery Education, TV Show parodies, etc.) to build understanding ● Use graphic organizers to guide notes, brainstorming, pre-writing, project planning, and test preparation ● Model through processes during assignments and elicit student-generated thoughts to determine gaps in understanding ● Highlight, bold, or underline main ideas in readings and in directions for writing assignments in the curricular areas ● Provide guiding questions to complete during the activity ● Provide chunking of instructional notes and activities to allow for formative assessment (checks for understanding) before moving on to the next stage ● Provide demonstrations, utilize pictures, or graphic to assist visual learners to support written text information ● Include oral discussions, oral presentations, group collaboration, or other oral delivery methods to support auditory learners ● Utilize hands-on activities, movement or rhythmic experiences to engage tactile/kinesthetic learners ● Provide chunking of assignments into manageable steps, including checklists that clarify directions for assignments ● Highlight distinctive features/key concepts ● Review, repeat, and clarify directions ● Allow for partial credit, when appropriate

Formative Assessments
- Sphero Bolt Introductory Activities   - BOLT Blocks 1: Roll Block Squares   - BOLT Blocks 2: Lights and Sounds   - BOLT Blocks 3: Matrix Emotions   - BOLT Blocks 4 : On Collision Event Pong   - BOLT Blocks 5: If Then Animal Toss   - BOLT Blocks 6: BOLT Sensor Storytelling Sidekick   - BOLT Blocks 7: Flashlight Function Tug o' War   - BOLT Blocks 8: Hot BOLTatoes and Variables   - vexCODE VR Computer Science Level 1: Block Coding Activities   - Introduction and Fundamentals   - Moving Your Robot   - Repeating Behaviors   - Navigating a Maze   - Detecting Walls from a Distance   - Knowing Your Location   - Decisions with Colors   - Moving Discs with Loops   - Developing Algorithms   - vexCODE VR Computer Science Level 1: Python Activities   - These are the same as the Block Coding Activities, but this time, students are not using any blocks to code. They are using Python (a common programming language) to code.

Summative Assessments
- Basic Block and Text Coding Assessment   - Basic Block and Text Coding Written Reflection

Performance Assessments
- Successfully code the robot to perform the desired task(s)   - Safely utilize tools and machines

RETURN TO CALENDAR

RETURN TO UNIT SUMMARIES

Unit 3:	Advanced Electronics, Components, and Circuitry Prototyping

Unit Summary
Now that students have some knowledge of basic electronics, circuitry, and some text-based coding under their belt, it is time to step things up a bit with the Arduino. The Arduino is a microcontroller, which means that it is a tool that is able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. Students will use the Arduino to learn yet another programming language, C++, and will create some really fun and unique projects, like a motorized pinwheel and a keyboard instrument. Students will use TinkerCAD to model, test, and refine their circuits before building the actual prototype, just like Electrical Engineers do on a daily basis.

Essential Questions	**Enduring Understandings**
• What is a microcontroller? • How do breadboards work? • What are some advantages/disadvantages of breadboarding with the Arduino? • What are some advantages/disadvantages of using TinkerCAD to model circuits before physically building them?	• Engineering design is a systematic and creative process that requires special tools and machinery to accomplish tasks. • Engineers will often model design solutions, circuits, and run simulations whenever they can to catch flaws in the designs before manufacturing and making their design. This often saves time, money and resources and can also be used to convey design ideas to clients.
• What is binary? • What is *true/false* and *high/low* in C++? • What are functions? • How are loops used to make code more efficient? • What is an *if/then* statement and how is it used?	• Programming languages provide variables, which are used to store and modify data. • A variety of control structures are used to change the flow of code, like sequences, loops, and conditionals.
• How do we know our code is written correctly? • If our code is written correctly, but the circuit is not working as intended, where else can we look for possible errors?	• Troubleshooting a problem is more effective when knowledge of the specific device along with a systematic process is used to identify the source of the problem. • Individuals develop programs using an iterative process involving design, implementation, testing and review.
• How are integers used as constants in code? • How can variables be used to control an LED or motor? • How are functions and algorithms related?	• Individuals evaluate and select algorithms based on performance, reusability, and ease of implementation. • Integers can be used as constants, which create variables, which can lead to advanced control of outputs, like LEDs and motors.
• Sometimes written code can be long and exhaustive. How can we manage such daunting tasks?	• Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose.

Behavioral Objectives/Learning Outcomes	Standards (NJSLS)
• Follow step by step directions to assemble a product or solve a problem, using appropriate tools to accomplish the task	8.2.3.ED.3
• Create and write code that uses clearly named variables to store and modify data • Create and write code that includes sequences, events, loops, and conditionals	8.1.5.AP.2-3
• Systematically apply troubleshooting strategies to identify and resolve hardware and software problems • Develop programs using an iterative process, implement the program design, and test the program to ensure it works as intended	8.2.8.ED.4 8.1.8.CS.4 8.1.5.AP.6
• Design algorithms to solve computational problems using a combination of original and existing algorithms • Create generalized computational solutions using collections instead of repeatedly using simple variables	8.1.12.AP.1-2
• Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects	8.1.12.AP.5

Interdisciplinary Connections
Computer Science, STEAM, Physics, Algebra, English

21st Century Skills
Critical thinking, communication skills, creativity, problem solving, perseverance, collaboration, information literacy, technology skills and digital literacy, media literacy, self-direction, social skills, literacy skills, innovations skills, thinking skills

Writing Assignments
This unit is very hands-on, but in the beginning, students will complete class notes on breadboarding, on the Arduino Board and any new component that has not been introduced in Unit 1: Basic Electronic Components. Students will complete these notes using fill-in-the blank Google Slides presentations. This will be printed out for students with slides on one side and a notes section on the other. Additionally, students will be “de-coding” each activity. For each activity, students will print the code they wrote and students will have to write out, line by line, what their code means and students will essentially be translating C++ to English.

Activities, Instructional Strategies, and Assignments

1. Students will start by downloading the Arduino software and getting familiar with the Arduino Uno board. There are many ports, plugs, and areas to attach other components and circuits on the Arduino. It is important that students know what all parts of the board are and do, because if used improperly or hooked up wrong, could damage the board permanently.
    a. Students will follow along with lectures, videos, and hands-on demonstrations
    b. The first few challenges will be heavily teacher-led and group-based, but students will have to hook up the circuits on their own using their individual boards and must submit their own written work (notes, reflections, etc.)
2. Arduino Activities - Students will be given their own Arduino Uno Starter Kit, complete with their own boards, collection of components, and Activity Book. The first few projects will be done in-depth with additional class notes, hands-on demonstrations and supplemental materials, like videos and hand-outs. Each of the activities will focus on new hardware and also new computer science concepts. Students will also be using TinkerCAD to model these circuits before using the software and hardware to make them a reality. At the beginning of each project, students will see the finished product, the completed code, and the entire circuit working as intended. From there, students will work backwards. For new hardware, the teacher will show students the component, provide videos and handouts, and show the component in a circuit. For new code, the teacher will show students the code, provide videos and handouts, write the code from scratch, and then have students explain, in a written document, what is happening in the code. New hardware and new code is introduced at the start of each new project in the student's Activity Books. See "Keyboard Instrument" below for an example:
    a. Spaceship Interface
    b. Love-o-Meter
    c. Color Mixing Lamp
    d. Mood Cue
    e. Light Theremin
    f. Keyboard Instrument
        i. Components: Piezo, resistor ladder (resistors in parallel)
        ii. Code: Arrays (of frequencies), sending analog values to the serial monitor, *if( )...else* statements, ranges, using &&
    g. Digital Hourglass
    h. Motorized Pinwheel
    i. Zoetrope
    j. Crystal Ball
    k. Knock Lock
    l. Touchy-Feely Lamp
    m. Tweak the Logo
    n. Hacking Buttons

Accommodations and Modifications (BHPRSD Accommodations and Modifications)

- Provide a variety of concrete examples from familiar contexts
- Build background knowledge of content and vocabulary from familiar contexts prior to readings
- Use mental models to building understanding through familiar contexts
- Provide oral & written instructions
- Incorporate multimedia/audio visual representation (YouTube, Discovery Education, TV Show parodies, etc.) to build understanding
- Model through processes during assignments and elicit student-generated thoughts to determine gaps in understanding
- Provide guiding questions to complete during the activity
- Provide chunking of instructional notes and activities to allow for formative assessment (checks for understanding) before moving on to the next stage

- Provide demonstrations, utilize pictures, or graphic to assist visual learners to support written text information
- Utilize hands-on activities, movement or rhythmic experiences to engage tactile/kinesthetic learners
- Provide chunking of assignments into manageable steps, including checklists that clarify directions for assignments
- Provide a clear, concise version of a scoring rubric prior to the assignment or assessment
- Highlight distinctive features/key concepts
- Provide choice of projects depending on the student's interests or strengths
- Review, repeat, and clarify directions
- Chunk sections of assessment
- Allow for partial credit, when appropriate
- Provide general assistance with organizational skills
- Keep rules simple and clear

Formative Assessments

- Breadboard Notes
- Arduino IDE Notes
- TinkerCAD Notes
- Arduino Activities
    - Spaceship Interface
    - Love-o-Meter
    - Color Mixing Lamp
    - Mood Cue
    - Light Theremin
    - Keyboard Instrument
    - Digital Hourglass
    - Motorized Pinwheel
    - Zoetrope
    - Crystal Ball
    - Knock Lock
    - Touchy-Feely Lamp
    - Tweak the Logo

Summative Assessments

- Final Arduino Activity: Hacking Buttons

Performance Assessments

- Successfully complete all Arduino Activities
    - Model in TInkerCAD
    - Breadboard and program circuits
- Safely utilize tools and machines

RETURN TO CALENDAR

RETURN TO UNIT SUMMARIES

Unit 4:	Soldering

Unit Summary
In this unit, students will learn how to safely solder, with particular emphasis on through-hole components on a circuit board. Students will learn to read schematic diagrams, identify components, and place these components accordingly on their Printed Circuit Board (PCP).

Essential Questions	Enduring Understandings
• What is solder? • What tools are needed to solder?	• Soldering requires special tools, like hot irons, sharp knives and cutters, and some specialized equipment, like helping hands and brass sponges.
• How do you safely solder? • How do you safely desolder a component from a PCB?	• Soldering has many steps, including some very important safety concerns, and these must always be followed.
• What is a schematic? • How are different components identified? • What kind of information is on a datasheet?	• Schematics help electrical engineers interpret a circuit and ensure that components are placed and soldered in the correct spot. • Some parts are identified by color, size, markings, or can be identified based on a datasheet.
• How does each component affect the circuit as a whole?	• In any system, each part serves a purpose. All parts must work correctly for the entire system to function.
• What can be done if a component or the circuit as a whole is not working properly? • How is desoldering different from soldering?	• Sometimes if a circuit is not working, it can be the fault of one single component. It is important to understand how each component works, how the circuit is supposed to behave, and to test each component, solder connection, and the integrity of the board itself.

Behavioral Objectives/Learning Targets	Standards (NJSLS)
• Define solder and describe what soldering is. • List the tools needed to solder.	8.2.2.ED.3 8.2.2.ITH.1-2
• Summarize the steps required to safely solder and always follow these steps when soldering.	8.2.3.AP.4
• Read and interpret schematic symbols and designs in order to create a fully functional circuit with all components correctly placed.	8.2.2.NT.1
• Successfully assemble and solder components on a circuit board. • Model and explain how the circuit works by identifying the relationship of each part and how all parts work together in the greater whole.	8.2.5.ED.3
• Troubleshoot and fix circuits or individual components that do not work as intended.	8.2.5.NT.1

Interdisciplinary Connections
STEAM, Physics, English

21st Century Skills
Critical thinking, communication skills, problem solving, perseverance, information literacy, technology skills, self-direction, social skills, literacy skills, thinking skills

Writing Assignments
Students will be given schematic diagrams and will be expected to read, interpret, and write a detailed explanation about the individual components and the circuit as a whole.

Activities, Instructional Strategies, and Assignments

This unit is project based and mostly self-paced.

1. Students will practice stripping and soldering stranded and solid wire.
2. Students will start soldering on practice boards to hone in their technique and ensure they know and follow soldering safety rules at all times.
3. Students will be given soldering kits with schematics and will assemble these kits based on the instructions manuals, schematics, and datasheets of components. For circuits that do not work as intended, students will troubleshoot and fix components to achieve a fully-functional, working circuit.

Accommodations and Modifications (BHPRSD Accommodations and Modifications)

- Provide general assistance with organizational skills
- Utilize homework assignment notebook/planner/agenda
- Provide written intermediate timelines for long assignments
- Have student monitor grade average
- Keep rules simple and clear
- Implement a behavior management system
- Build background knowledge of content and vocabulary from familiar contexts prior to readings
- Use mental models to building understanding through familiar contexts
- Provide oral & written instructions
- Incorporate multimedia/audio visual representation (YouTube, Discovery Education, TV Show parodies, etc.) to build understanding
- Model through processes during assignments and elicit student-generated thoughts to determine gaps in understanding
- Provide guiding questions to complete during the activity.
- Provide chunking of instructional notes and activities to allow for formative assessment (checks for understanding) before moving on to the next stage.
- Provide demonstrations, utilize pictures, or graphic to assist visual learners to support written text information
- Utilize hands-on activities, movement or rhythmic experiences to engage tactile/kinesthetic learners
- Provide chunking of assignments into manageable steps, including checklists that clarify directions for assignments
- Highlight distinctive features/key concepts
- Provide choice of projects depending on the student's interests or strengths
- Review, repeat, and clarify directions
- Provide general assistance with organizational skills
- Keep rules simple and clear

Formative Assessments
• Wire stripping practice • Soldering motors practice • Soldering practice board • Safety checks • Build checkpoints

Summative Assessments
• Final circuit • Final written analysis of circuit

Performance Assessments
• Build a fully functional and working circuit • Safely utilize tools and machines

RETURN TO CALENDAR

RETURN TO UNIT SUMMARIES

Unit 5:	Additive Manufacturing

Unit Summary
Once students have successfully soldered their circuits, it's time to create a custom housing assembly for their circuits to protect it from damage. Students will take measurements using a micrometer, learn about fastener sizing, pitch, and threads, and will create a multi-part assembly to encase their projects. Once printed and assembled, students will share their designs with the class.

Essential Questions	**Enduring Understandings**
• What is additive manufacturing? • How is additive manufacturing similar/different from traditional manufacturing techniques? • Is 3D printing sustainable and/or eco-friendly?	• Additive manufacturing uses data from CAD software to direct hardware to deposit material, layer upon layer, in precise geometric shapes. As its name implies, additive manufacturing adds material to create an object. By contrast, when creating an object by traditional means, it is often necessary to remove material through milling, machining, carving, shaping, etc. and this often produces a lot of waste material.
• What information is required on an Engineering drawing? • What is the purpose of detail and section views?	• Three-view Engineering drawings must have the front, top and side views, and dimensions for each piece of geometry, part or sub-assembly. • Detail and section views are used to show smaller geometries, parts, holes, extrusions, etc. that require a closer look or are hidden by other part faces.
• What kinds of screws and fasteners will work best?	• Engineering design requirements and specifications involve making trade-offs between competing requirements and desired design features.
• How do we evaluate a design's effectiveness?	• When designs are evaluated, we must consider how well that solution meets requirements and compare requirements, specifications and constraints.

Behavioral Objectives/Learning Targets	Standards (NJSLS)
• Consider and compare the sustainability of additive manufacturing and more specifically, 3D printing with ABS plastics.	8.2.5.ETW.4-5 8.2.8.ITH.2-4
• Create scaled and dimensioned Engineering drawings for the custom enclosure design, complete with part files for each part, an assembly, and detail/section view drawings as necessary.	8.2.8.ED.3 8.2.12.ED.2
• Evaluate various screws, nuts, and other fasteners to determine which is best for this specific design. • Consider screw length, thickness, pitch, pricing, availability, etc. when making decisions about which fastener/hardware to use.	8.2.8.ED.5-7 8.2.12.ED.3
• Evaluate the effectiveness of the enclosure design based on factors that are related to its requirements, specifications, and constraints.	8.2.12.ED.5

Interdisciplinary Connections
STEAM, Geometry, Physics, English

21st Century Skills
Critical thinking, communication skills, creativity, problem solving, perseverance, collaboration, information literacy, technology skills and digital literacy, media literacy, self-direction, social skills, literacy skills, innovations skills, thinking skills

Writing Assignments
Students will write a detailed explanation about their design, hinge, fastener, and closing mechanism choices, as well as other design considerations. Students will give a short presentation to the class summarizing their design considerations from this assignment.

Activities, Instructional Strategies, and Assignments

1. Students will learn to read Dial Calipers with a hands-on, in-class assignment where they will measure real-world objects and compare their answers with others in the class.
2. Students will learn about a variety of fasteners, with particular attention to screws. Students will learn to ID different types of fasteners, how they are measured in length, diameter, thread count, pitch, etc. Students will complete a hands-on activity where they will have to identify certain characteristics about a random assortment of screws.
3. Students will all be given the same box/enclosure to house their circuit boards and will use the dial calipers to re-create this box/enclosure in a CAD program.
4. Students will then meet with a partner and discuss how they can redesign this enclosure to make it better fit the criteria/constraints.
5. Students will then redesign the box/enclosure and 3D print their new enclosures and fully assemble this using the necessary fasteners.
6. Students will complete a final written reflection detailing design choices and give a short presentation to the class that summarizes their progression through this unit and design choices made.

Accommodations and Modifications (BHPRSD Accommodations and Modifications)

- Provide a variety of concrete examples from familiar contexts
- Build background knowledge of content and vocabulary from familiar contexts prior to readings
- Use mental models to building understanding through familiar contexts
- Provide oral & written instructions
- Incorporate multimedia/audio visual representation (YouTube, Discovery Education, TV Show parodies, etc.) to build understanding
- Use graphic organizers to guide notes, brainstorming, pre-writing, project planning, and test preparation
- Model through processes during assignments and elicit student-generated thoughts to determine gaps in understanding
- Highlight, bold, or underline main ideas in readings and in directions for writing assignments in the curricular areas
- Provide guiding questions to complete during the activity
- Provide chunking of instructional notes and activities to allow for formative assessment (checks for understanding) before moving on to the next stage
- Choose cooperative learning groups to ensure effective work, maximize productivity and support socialization
- Provide demonstrations, utilize pictures, or graphic to assist visual learners to support written text information
- Include oral discussions, oral presentations, group collaboration, or other oral delivery methods to support auditory learners
- Utilize hands-on activities, movement or rhythmic experiences to engage tactile/kinesthetic learners
- Provide chunking of assignments into manageable steps, including checklists that clarify directions for assignments
- Provide a clear, concise version of a scoring rubric prior to the assignment or assessment
- Highlight distinctive features/key concepts
- Provide choice of projects depending on the student's interests or strengths
- Review, repeat, and clarify directions
- Chunk sections of assessment

Formative Assessments
- How to Read Dial Calipers assignment   - Fastener Identification assignment   3D Model a Screw assignment   - Enclosure Sketches and Brainstorming   - Pair/Share with Partner 1-2 times during build/design

Summative Assessments
- Final working enclosure, 3D printed, with all hardware and fasteners attached   - Final written reflection detailing design choices

Performance Assessments
- Create a 3D printed enclosure for their circuits that meets all criteria and constraints   - Safely utilize tools and machines

RETURN TO CALENDAR

RETURN TO UNIT SUMMARIES

Unit 6:	Engineering Design Process (District Tech Challenge)

Unit Summary
In this unit, students will work collaboratively in small teams to solve an open-ended design challenge using the steps of the Engineering Design Process (EDP). In lower level Design & Tech classes, students are walked through each step of the process at a slower and more in-depth pace. But in higher level classes, like Robotics, students are free to follow the steps of the TDP at their own pace and must document their progression through the TDP by means of a Design Journal. Once students have created a viable solution, they will refine, retest, and rebuild their design in hopes of being crowned the district champion as they compete against other teams from the Technology Education classes in the district.

Essential Questions	**Enduring Understandings**
• What are the steps of the engineering design process? • Why is it important that the steps are done in order?	• There are many steps of the engineering design process. Students should consider the importance of following the steps in order. For instance, knowing the entire scope of the project, and materials and budget is necessary, before building the physical product or solution.
• How can collaboration help/hinder a design? • Why do Engineers work on teams with others from different disciplines?	• Recognize that collaboration is a great thing, but can be challenging at times. It is important that all group members contribute and feel respected and heard. • Collaborating with others to brainstorm different solutions and synthesize many ideas into the best solution will often prove more successful than just one mind alone.
• What are criteria and constraints and how do these affect designs?	• Every design challenge has its own criteria and constraints. It is crucial that students know what these are and ensure their design meets them.
• How do you safely use tools and machines?	• Safety is of the utmost importance in the Design & Tech classroom. Students must always use tools and machines correctly and safely to create any design solution/product.
• Why is it important to document progress through the design process?	• Documenting progress is critical because it provides a history of a student's progress and shows how their ideas became a reality.
• What are Engineering drawings? • Why are Engineering drawings used in product design?	• Engineering drawings are useful because they show dimensions, assemblies, and provide all information needed to make a product. WIthout these, products could not be made accurately.
• Why is it important to reflect upon a design? • How do Engineers use reflection to redesign and make products better?	• Once a product is made, the process is not over. The design process is iterative. The design must be critiqued again and, in many cases, redesigned to better solve the design challenge.

Behavioral Objectives/Learning Targets	Standards (NJSLS)
• Identify and apply the steps of the design process to create a solution that best solves the Tech Challenge.	8.2.8.ED.2
• Collaborate with others to collect information, brainstorm to solve a problem, and evaluate all possible solutions to provide the best results with supporting sketches and models.	8.2.2.ED.2 8.2.5.ED.2
• Identify constraints and limitations in this design challenge and ensure that the final product adheres to these constraints and limitations.	8.2.2.ED.4
• Follow all lab, tool, and machine safety rules.	8.2.2.ED.3
• Create, develop, and maintain a Design Journal that documents progress through the design challenge, including decisions made as a result of specific constraints and trade-offs.	8.2.5.ED.5 8.2.8.ED.7
• Create a scaled, dimensioned, and accurate Engineering drawings using a CAD-based program, like Onshape.	8.2.8.ED.3 8.2.12.ED.2
• Analyze the effectiveness of the design and redesign, as necessary. • Evaluate how well the design has met its intended purpose and identify any shortcomings it might have.	8.2.5.ITH.2-3 8.2.8.ED.2 8.2.12.ED.6

Interdisciplinary Connections
STEAM, Geometry, Physics, English

21st Century Skills
Critical thinking, communication skills, creativity, problem solving, perseverance, collaboration, self-direction, social skills, innovation skills, thinking skills

Writing Assignments
Students are expected to keep and maintain a detailed Design Journal of their progress through the TDP. Each step of the TDP has its own section in the journal starting with Identifying the Problem and Understanding the Design Brief. Once the Tech Challenge is over, students will write an essay and reflect upon their experience.

Activities, Instructional Strategies, and Assignments

1. Students will read through the Design Brief, which will include all of the information related to this year's Tech Challenge, including things like: rules, constraints, materials, timeline, rubric, etc.
2. Students will be given a Design Journal Template, which students will complete throughout the challenge.
    a. Students complete a Daily Activity Log detailing the work accomplished that day.
    b. Students begin by reading through the Design Brief and breaking it down into smaller chunks and focusing on the most important details, rules, and regulations, with extra emphasis on the Criteria and Constraints section.
    c. Students then Research and Brainstorm some ideas and Sketch out a bunch of different ideas, which will serve as a good jumping off point. These ideas are constantly assessed and students decide what parts might work, what parts don't, and they might even combine ideas from one sketch to another.
    d. Students then pick one final design, which is typically a synthesis of their early sketches.
    e. Students then meet with their group and together, these group members pull all of their ideas together into one idea and submit a Detailed Sketch with a Design Proposal.
3. Once approved, students get materials and start building the physical prototype.
4. Once a solution is made, students test, revise, and adapt their designs to ensure that their final prototype is the BEST solution for the given design challenge.
5. Students compete against others in the class and then, if they are the best in the class, move on to compete against others in the district in hopes of being crowned the "District Champion".
6. After the competition, students reflect on their design's effectiveness and their progression through the design process and submit a final Reflection Essay.
7. Students then take their projects and create a CAD-based 3D model of their final prototype. In addition to this, students create a series of drawings, including fully dimensioned and annotated 3-view drawings for each part, an assembly, and a materials list.

Accommodations and Modifications (BHPRSD Accommodations and Modifications)
● Provide oral & written instructions ● Incorporate multimedia/audio visual representation (YouTube, Discovery Education, TV Show parodies, etc.) to build understanding ● Model through processes during assignments and elicit student-generated thoughts to determine gaps in understanding ● Provide guiding questions to complete during the activity ● Provide chunking of instructional notes and activities to allow for formative assessment (checks for understanding) before moving on to the next stage ● Choose cooperative learning groups to ensure effective work, maximize productivity and support socialization ● Use multiple intelligences or the student's learning style to facilitate effective learning when a student is having difficulty grasping concepts ● Include oral discussions, oral presentations, group collaboration, or other oral delivery methods to support auditory learners ● Utilize hands-on activities, movement or rhythmic experiences to engage tactile/kinesthetic learners ● Provide chunking of assignments into manageable steps, including checklists that clarify directions for assignments ● Provide a clear, concise version of a scoring rubric prior to the assignment or assessment ● Highlight distinctive features/key concepts ● Review, repeat, and clarify directions ● Chunk sections of assessment ● Provide written intermediate timelines for long assignments ● Keep rules simple and clear

Formative Assessments
● Daily Activity Log ● Sketches for Team Logo ● Final Team Logo ● Sketches for Journal Cover Design ● Final Journal Cover Design ● Thumbnail Sketch Ideas ● Detailed Sketch of Best Idea w/ Written Summary and Build Proposal ● Parts List/Bill of Materials ● Build “Checkpoints” (i.e. add wheels to project, get a working release mechanism, etc.)

Summative Assessments
● Final drawing (completed after project is built) using CAD-based programs, like Onshape ● Final evaluation of the physical built solution, based on a rubric ● Final evaluation of the Design Journal, based on a rubric ● Essay reflecting on the entire Tech Challenge and progression through the TDP

Performance Assessments
● Create and build a physical solution to the Tech Challenge ● Safely utilize tools and machines

RETURN TO CALENDAR

RETURN TO UNIT SUMMARIES

Unit 7:	Advanced Text Coding for Robotic Control

Unit Summary
Students will build and program a Tetrix robot to perform autonomous tasks using C++ within the Arduino IDE software. These tasks will start out fairly simple - Drive forward. Stop. Turn right. But as students progress through, they will add more hardware and sensors to their robots and the tasks will get more and more complex - Use the Ultrasonic Sensor to sense obstacles to avoid them, use the Line Finder sensor to get your robot to parallel park, or combine the two to have a fully autonomous robot successfully navigate itself through a maze.

Essential Questions	Enduring Understandings
● How is coding with PRIZM similar/different from coding with the Arduino? ● What is Tetrix PRIZM and how is it used to control outputs? ● How does the line finder and ultrasonic sensors work? ● How are DC motors and servo motors different/similar?	● Engineering design is a systematic and creative process that requires special tools and machinery to accomplish tasks ● PRIZM and Arduino are two microcontrollers used to control a variety of outputs. ● Sensors like the line finding and ultrasonic sensor are used to control outputs. ● DCs have high RPM and are continuous rotation (C.R.), whereas servos have more torque and not all are C.R.
● How is it possible that two different students can have two different code instructions, yet their robots both successfully accomplish the task at hand?	● Engineering design evaluation, a process for determining how well a solution meets requirements, involves systematic comparisons between requirements, specifications and constraints. ● There are many ways to code a robot to successfully achieve a task or solve a problem.
● How can functions be used within other called functions? ● What is the difference between conditional and comparison statements? ● Why are comparison statements used with sensor data? ● How are *if/then statements* used in C-based languages? ● How are *if/else statements* used in C-based languages?	● A function is a “chunk” of code. This chunk can be nested in other chunks to improve the code written. ● Conditionals run only under certain conditions, if certain requirements are met. Comparisons run when we want to control the program flow. ● Comparison statements allow us to gather input from sensor data, to see if a value is true (1) or false (0) and then have our robots behave one way or the other. ● If it is raining out, then you will open and use your umbrella. If it is sunny out, you will put on your sunglasses, but you won’t need your umbrella. The same logic is applied to coding and can help us control robots.
● How are algorithms used in C-based languages to make our code more efficient and provide advanced robotic control?	● Algorithms are a group of functions that run on a designated range of elements. Algorithms are all around us: a recipe for baking a cake or doing a load of laundry. ● When adding additional motors and sensors, algorithms help us control the robot more methodically.
● What if our code is correct, but the robot is not responding as expected? ● How can we find, edit, and fix	● Writing code is an iterative process that requires constant implementation, testing and review. ● When troubleshooting robots, we have to look at the written code as well as the inputs, outputs, and even the

errors within our code to ensure we are successful?
● Why might it be beneficial to break a large task down into smaller, sub-tasks? ● How do the different sub-assemblies contribute to the overall function of the robot?

physical structures that make up the robot we are controlling - and maybe even the controller itself.
● Code can be broken down into smaller parts to facilitate design, implementation, review, and can help aid in finding bugs and errors. ● Building a robot requires specific attention to each individual sub-assembly and lots of care and attention must be given to the smallest of parts.

Behavioral Objectives/Learning Outcomes	Standards (NJSLS)
• Follow step by step directions to build theTaskBot. • Follow along with online modules and apply learned content to aid in coding the TaskBot using Arduino IDE.	8.2.3.ED.3
• Evaluate the effectiveness and efficiency of their written code. • Evaluate the effectiveness of the robot's physical design, based on requirements, specifications and constraints	8.2.12.ED.5
• Create and write code that uses clearly named variables to store and modify data • Create and write code that includes sequences, events, loops, and conditionals	8.1.5.AP.2-3
• Design algorithms to solve computational problems using a combination of original and existing algorithms • Create generalized computational solutions using collections instead of repeatedly using simple variables	8.1.12.AP.1-2
• Systematically apply troubleshooting strategies to identify and resolve hardware and software problems	8.2.8.ED.4 8.1.8.CS.4
• Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects	8.1.12.AP.5

Interdisciplinary Connections
Computer Science, STEAM, Physics, Algebra, English

21st Century Skills
Critical thinking, communication skills, creativity, problem solving, perseverance, collaboration, information literacy, technology skills and digital literacy, media literacy, global awareness, self-direction, social skills, literacy skills, social responsibility, innovations skills, thinking skills

Writing Assignments
Students will complete a series of notes using fill-in-the blank Google Slides presentations. This will be printed out for students with slides on one side and a notes section on the other. Students will be expected to maintain forward progress on the Tetrix VR modules, some of which require input, writing, and reflection, in addition to the written code itself. After successful completion of each activity, students will submit screenshots of their code, videos of their robot performing specific tasks, or will have to submit a Google Form Quiz, or answer a series of questions on a Google Doc or Google Slides Presentation and submit to Google Classroom as entrance and/or exit tickets.

Activities, Instructional Strategies, and Assignments

This unit is largely self-paced. The progressive nature of the activities enables robotic creations to come to life quickly and easily, meaning students can experience instant success and focus more classroom time on problem-solving and applying their STEM knowledge. These activities offer students the following:

- Introduction
- List of Parts Needed
- A “Building the Knowledge Base” Section detailing background information needed for the activity
- A “Moving Forward” section that helps summarize newly learned material
- A “Real World” and “STEM” Connections section tying the content to the real world
- An additional “Hacking the Code” Activity for students who want to challenge themselves and take things a step farther

For each of these activities, students will follow along with lectures, videos and demonstrations, review the new content as a class, and then code, build, and experiment on their robots. At the start of each activity, students will log on to the Tetrix Virtual Robotics App and begin learning about the activity from this interface. That means that students will be moving seamlessly between virtually coding within the simulator and testing code on their real-world robots. They will iterate in both the digital and physical world, just like real engineers. After a student has completed the VR module, they will transition to their real-world robot and will build or add components and then write the code to control their bots. After successful completion of the activity in the real world, students will then submit screenshots of their code, videos of their robot performing specific tasks, or will have to submit a Google Form Quiz, or answer a series of questions on a Google Doc or Google Slides Presentation and submit to Google Classroom as entrance and/or exit tickets.

The Tetrix Programming Guide Activities are as follows:

1. Getting Started Activities
    a. Hello World!
    b. Moving Your DC Motors
    c. Moving Your Servo Motors
    d. Introduction to the Line Find Sensor (Save for Activity for “Drive to a Line and Stop”)
    e. Introduction to the Ultrasonic Sensor (Save for Activity “Avoiding Obstacles”)
2. Build the TaskBot
3. TaskBot Activities
    a. Drive Forward
    b. Drive in a Circle
    c. Drive in a Square
    d. Simplify the Square
    e. Drive to a Line and Stop
    f. Follow a Line
    g. Drive Toward a Wall and Stop
    h. Avoiding Obstacles
    i. Combining the Sensors

Accommodations and Modifications (BHPRSD Accommodations and Modifications)
• Provide a variety of concrete examples from familiar contexts • Build background knowledge of content and vocabulary from familiar contexts prior to readings • Use mental models to building understanding through familiar contexts • Provide oral & written instructions • Incorporate multimedia/audio visual representation (YouTube, Discovery Education, TV Show parodies, etc.) to build understanding • Use graphic organizers to guide notes, brainstorming, pre-writing, project planning, and test preparation • Model through processes during assignments and elicit student-generated thoughts to determine gaps in understanding • Highlight, bold, or underline main ideas in readings and in directions for writing assignments in the curricular areas • Provide guiding questions to complete during the activity • Provide chunking of instructional notes and activities to allow for formative assessment (checks for understanding) before moving on to the next stage • Choose cooperative learning groups to ensure effective work, maximize productivity and support socialization • Use multiple intelligences or the student's learning style to facilitate effective learning when a student is having difficulty grasping concepts • Provide demonstrations, utilize pictures, or graphic to assist visual learners to support written text information • Include oral discussions, oral presentations, group collaboration, or other oral delivery methods to support auditory learners • Utilize hands-on activities, movement or rhythmic experiences to engage tactile/kinesthetic learners • Provide chunking of assignments into manageable steps, including checklists that clarify directions for assignments • Provide a clear, concise version of a scoring rubric prior to the assignment or assessment • Highlight distinctive features/key concepts • Provide choice of projects depending on the student's interests or strengths • Provide peer assistance/study groups • Review, repeat, and clarify directions • Chunk sections of assessment • Allow for partial credit, when appropriate • Allow use of familiar contexts to demonstrate understanding of key concepts when use of text evidence is not necessary • Provide general assistance with organizational skills • Provide written intermediate timelines for long assignments • Have student monitor grade average • Keep rules simple and clear

Formative Assessments
● TaskBot Activities - including VR module, physical build, and reflections/analysis assignments ○ Drive Forward ○ Drive in a Circle ○ Drive in a Square ○ Simplify the Square ○ Drive to a Line and Stop ○ Follow a Line ○ Drive Toward a Wall and Stop ○ Avoiding Obstacles

Summative Assessments
● Final Taskbot Activity: Combining the Sensors

Performance Assessments
● Successfully complete all Tetrix TaskBot Activities ○ Complete each module ○ Build, code, and complete each real-world activity ● Safely utilize tools and machines

RETURN TO CALENDAR

RETURN TO UNIT SUMMARIES

Unit 8:	Robotics Design

Unit Summary
In this unit, students will work in teams to synthesize everything they learned throughout the year and will be given open-ended design challenges where they will compete against others in the class. Some challenges will require additional sensors and circuitry, some might require a custom fabricated part, some might require Remote Control (RC) and Bluetooth capabilities. The possibilities are endless!

Essential Questions	Enduring Understandings
• What kinds of screws and fasteners will work best for the various sub-assemblies?	• Engineering design requirements and specifications involve making trade-offs between competing requirements and desired design features.
• How are assemblies and subassemblies depicted in Three-View drawings? • When would a detail or section view be appropriate to include in a set of drawings?	• Three-view Engineering drawings must have the front, top and side views, and dimensions for each piece of geometry, part or sub-assembly. • Detail and section views are used to show smaller geometries, parts, holes, extrusions, etc. that require a closer look or are hidden by other part faces.
• How can collaboration help/hinder a design? • Why do Engineers work on teams with others from different disciplines?	• Engineers use science, mathematics, and other disciplines to improve technology. Increased collaboration among engineers, scientists, and mathematicians can improve their work and designs. • Complex programs are developed, tested, and analyzed by teams drawing on the members' diverse strengths and using a variety of resources, libraries, and tools.
• How can things like variables, loops and conditionals be used to control a robot to perform complex tasks?	• Programming languages provide variables, which are used to store and modify data. • A variety of control structures are used to change the flow of code, like sequences, loops, and conditionals.
• How do we know our code is written correctly? • If our code is written correctly, but the circuit is not working as intended, where else can we look for possible errors?	• Troubleshooting a problem is more effective when knowledge of the specific device along with a systematic process is used to identify the source of the problem. • Individuals develop programs using an iterative process involving design, implementation, testing and review.
• How are algorithms evaluated?	• Individuals evaluate and select algorithms based on performance, reusability, and ease of implementation.
• Sometimes written code can be long and exhaustive. How can we manage such daunting tasks?	• Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose.

Behavioral Objectives/Learning Outcomes	Standards (NJSLS)
• Evaluate various screws, nuts, and other fasteners to determine which is best for this specific design • Consider screw length, thickness, pitch, pricing, availability, etc. when making decisions about which fastener/hardware to use	8.2.8.ED.5-7 8.2.12.ED.3
• Create scaled engineering drawings for a new product or system and make modifications to increase optimization based on feedback	8.2.12.ED.2
• Explain how different groups can contribute to the overall design of a product • Redesign an Existing product to improve its form or function	8.2.12.NT.1-2
• Create and write code that uses clearly named variables to store and modify data • Create and write code that includes sequences, events, loops, and conditionals	8.1.5.AP.2-3
• Systematically apply troubleshooting strategies to identify and resolve hardware and software problems	8.2.8.ED.4 8.1.8.CS.4 8.1.8.AP.6-9
• Design algorithms to solve computational problems using a combination of original and existing algorithms • Create generalized computational solutions using collections instead of repeatedly using simple variables	8.1.12.AP.1-2 8.1.8.AP.2-3
• Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects	8.1.12.AP.5
• Collaboratively document and present design decisions in the development of complex programs	8.1.12.AP.9

Interdisciplinary Connections
Computer Science, STEAM, Physics, Algebra, English

21st Century Skills
Critical thinking, communication skills, creativity, problem solving, perseverance, collaboration, information literacy, technology skills and digital literacy, media literacy, global awareness, self-direction, social skills, literacy skills, social responsibility, innovations skills, thinking skills

Writing Assignments
- Each challenge will require students to write original, unique code using C-based computer programming languages   - For each challenge, there will be: - Build Proposal Identifican and Brainstorming Written Activity   - Sketch of Design Proposals   - Pair up with partner halfway through the build process and assess one another's designs   - Then, at the end of each challenge, students will write and reflect upon how well their design met the Design Proposal and will have to speak directly to elements of their Build Proposal Identification/Brainstorming, early Sketches, various iterations, what was talked about, learned, and revised in the partner meet-up, and the final prototype   - At the end of each challenge, students will meet with their groups and talk about their designs, from brainstorming to the final prototype. After writing an individual reflection, students will meet with their team members and give a short oral presentation detailing what was learned, problems faced and solutions made, etc.

Activities, Instructional Strategies, and Assignments

This unit is largely self-paced and heavily cooperative. Students will work in teams to compete against others in the class as they further develop their critical thinking and problem solving skills through a series of open-ended design challenges.

In general each activity will be split into three main parts:

1. Design and Build a Prototype. Example Activity: Design a garbage-moving robot that can push golf balls into a collection area
2. Test and Analyze Prototype.
3. Redesign and Improve the Prototype. Example Activity: Add an actuator to the robot to improve functionality.

Some challenges might be:

- Golf balls represent toxic materials. Ping-Pong balls represent recyclable materials. These will need to avoid contamination. Students will have a limited amount of time to “Sort the Trash” and clean up the landfill.
- Warehouse bot operations:
    - Navigation Systems: experiment with drive trains, gear trains and motor controllers.
    - Lift Systems: design a system to lift as much weight as possible.
    - Grippers and Scoops: experiment with different grippers and scoops to see how much weight they can hold.
    - Combine all three systems to create a Warehouse Bot: Students will have a limited amount of time to place empty spools in storage bins or stack them in the stacking area.
- Autonomous Rescue Robots: Students will convert their Warehouse Robots into Autonomous Bots using controllers and sensors from the previous unit.
    - Students will start by adding the line finding and ultrasonic sensor to their bots and ensure that the robot can follow a line as well as avoid obstacles.
    - Students write original code to get their robots to make sharp turns when following a line or sensing an object.
    - Students will then write an original program where the robot detects a wall and determines the best way to go around an obstacle.

For the latter two competitions, students will be given Build Proposals that detail Criteria and Constraints of their robot and their code. Students will be assessed on how well their design meets these requirements and students will be reflecting on how well their project performed, but also how well their design met the original Build Proposals. Students will give multimedia presentations on the effectiveness of their designs and present their findings to their class.

Students will spend the last week of the unit cleaning up the competition area, breaking down robots, organizing parts, and prepping their kits for next year’s students.

Accommodations and Modifications (BHPRSD Accommodations and Modifications)
• Provide a variety of concrete examples from familiar contexts • Build background knowledge of content and vocabulary from familiar contexts prior to readings • Use mental models to building understanding through familiar contexts • Provide oral & written instructions • Incorporate multimedia/audio visual representation (YouTube, Discovery Education, TV Show parodies, etc.) to build understanding • Use graphic organizers to guide notes, brainstorming, pre-writing, project planning, and test preparation • Model through processes during assignments and elicit student-generated thoughts to determine gaps in understanding • Highlight, bold, or underline main ideas in readings and in directions for writing assignments in the curricular areas • Provide guiding questions to complete during the activity • Provide chunking of instructional notes and activities to allow for formative assessment (checks for understanding) before moving on to the next stage • Choose cooperative learning groups to ensure effective work, maximize productivity and support socialization • Use multiple intelligences or the student's learning style to facilitate effective learning when a student is having difficulty grasping concepts • Provide demonstrations, utilize pictures, or graphic to assist visual learners to support written text information • Include oral discussions, oral presentations, group collaboration, or other oral delivery methods to support auditory learners • Utilize hands-on activities, movement or rhythmic experiences to engage tactile/kinesthetic learners • Provide chunking of assignments into manageable steps, including checklists that clarify directions for assignments • Provide a clear, concise version of a scoring rubric prior to the assignment or assessment • Highlight distinctive features/key concepts • Provide choice of projects depending on the student’s interests or strengths • Provide peer assistance/study groups • Review, repeat, and clarify directions • Chunk sections of assessment • Allow for partial credit, when appropriate • Allow use of familiar contexts to demonstrate understanding of key concepts when use of text evidence is not necessary • Provide general assistance with organizational skills • Provide written intermediate timelines for long assignments • Have student monitor grade average • Keep rules simple and clear

Formative Assessments
- For each challenge, there will be: - Build Proposal Identifican and Brainstorming Written Activity   - Sketch of Design Proposals   - Pair up with partner halfway through the build process and assess one another's designs   - Then, at the end of each challenge, students will write and reflect upon how well their design met the Design Proposal and will have to speak directly to elements of their Build Proposal Identification/Brainstorming, early Sketches, various iterations, what was talked about, learned, and revised in the partner meet-up, and the final prototype   - Build checkpoints   - Landfill Clean-Up Challenge   - Ultrasonic and Line Find Design Challenge   - Hardware (Bucket/Scoop/Shovel) Design Challenge   - Warehouse Bot Design Challenge   - Navigation Systems   - Lift Systems   - Gripper/Scoop Systems

Summative Assessments
- Autonomous Rescue Robot Design Challenge   - Autonomous Rescue Robot Group Reflection Presentation   - Autonomous Rescue Robot Individual Reflection Paper

Performance Assessments
- Successfully complete all Tetrix TaskBot Activities   - Build   - Code   - Test   - Safely utilize tools and machines

[RETURN TO CALENDAR]

[RETURN TO UNIT SUMMARIES]