Level 1 Robotics (Middle School & HS) Standards Alignment

CCSS Mathematics Alignment for Robotics

Mathematical Practice	Application in Curriculum
MP1 Make sense of problems and persevere in solving them	Chapters are all based around solving real-world robot problems; students must make sense of the problems to inform their solutions
MP2 Reason abstractly and quantitatively	Programming requires students to reason about physical quantities in the world to plan a solution, then calculate or estimate them for the robot
MP4 Model with mathematics	Many processes, including the process of programming itself, must be systematically modeled on both explicit and implicit levels
MP6 Attend to precision	Robots require precise (and accurate) input, or their output action will be correspondingly sloppy
MP7 Look for and make use of structure	Understanding the structure of the physical environment, the interrelated components of robot hardware and software, and commands within a program are vital to successful solutions
MP8 Look for and express regularity in repeated reasoning	Any programmed solution to a class of problems relies on the programmer recognizing and exploiting important patterns in the problem structure. There is also an emphasis throughout the course on recognizing common programmatic patterns, as well as patterns within a solution that invite the use of Loops.

CCSS Mathematics Alignment for Robotics

Standard (CCSS Math Content)	Application in Curriculum
6.RP.A.1 Understand the concept of a ratio and use ratio language to describe a ratio relationship between two quantities	Students use ratio language to describe and make use of the relationship between quantities such as Inputted Time and Distance Traveled
6.RP.A.2 Understand the concept of a unit rate a/b associated with a ratio a:b with b!=0, and use rate language in the context of a ratio relationship	The relationship between Inputted Time and Distance Traveled is a rate, customarily understood through a unit rate such as "# cm per second"
6.R.A.3 Use ratio and rate reasoning to solve real-world and mathematical problems	Students are required to apply ratios and rates when they build their prototype examples of their real-world robots
7.RP.A.3 Use proportional relationships to solve multistep ratio and percent problems	Comparisons between rate-derived quantities

CCSS English Language Arts Alignment for Robotics

Standard (CCSS ELA-Literacy)	Application in Curriculum
WHST.6-8.1 Write arguments focused on discipline-specific content [See also: WHST.6-8.1.a to WHST.6-8.1.e]	Reflection Questions ask students to analyze, evaluate, and synthesize arguments in response to robotics and programming problems
WHST.6-8.4 Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.	Reflection Question tasks include composing technical critiques, technical recommendations, and creative synthesis.

NGSS Alignment for Robotics

Standard	Application in Curriculum
MS-ETS1-2. Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.	Solving challenges requires students to create and evaluate both hardware and software designs according to scenario scoring criteria. Some Reflection Questions require students to make recommendations between competing alternatives based on criteria that they define.
MS-ETS1-4. Develop a model to generate data for iterative testing and modification of a proposed object, tool, or process such that an optimal design can be achieved.	When solving more difficult and complex challenges, students are guided toward iterative testing and refinement processes. Students must optimize program parameters and design.
HS-ETS1-2. Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.	Problem Solving methodology for challenges directs students to break down large problems into smaller solvable ones, and build solutions up accordingly; challenges give students opportunities to practice, each of which is based on a real-world robot
HS-ETS1-3. Evaluate a solution to a complex real-world problem based on prioritized criteria and trade-offs that account for a range of constraints, including cost, safety, reliability, and aesthetics as well as possible social, cultural, and environmental impacts.	Some Reflection Questions require students to make recommendations about real-world policies (e.g. requiring sensors on automobiles) based on the impact of that decision

CSP Alignment for Robotics

Learning Objective	Application in Curriculum
1.1.1 Use computing tools and techniques to create artifacts. [P2]	Challenge activities result in the creation of a (simple) algorithmic solution and an accompanying program that implements it.
1.1.2 Collaborate in the creation of computational artifacts. [P6]	Students work in teams to accomplish tasks.
1.1.3 Analyze computational artifacts. [P4]	Students perform debugging on their own code, as well as analyze and evaluate others' code and suggested code in Reflection Questions.

1.3.1 Use programming as a creative tool. [P2]	Students use programming to solve model challenges based on challenges real robots face.
2.2.1 Develop an abstraction. [P2]	Robots gather information about the world through sensors, which turn physical qualities of the world into digital abstractions. Students must understand and work with this data to develop then implement their solution algorithms.
2.3.1 Use models and simulations to raise and answer questions. [P3]	Students construct and use a "program flow" model of programming itself to understand how the robot uses data to make decisions and control the flow of its own commands.
4.1.1 Develop an algorithm designed to be implemented to run on a computer. [P2]	Students develop solution algorithms to each challenge and mini-challenge problem before implementing them as code. Reflection Questions also ask students to evaluate algorithms expressed as pseudocode.
4.2.1 Express an algorithm in a language. [P5]	Students develop code to robotics challenges in the Python programming language.
5.1.1 Explain how programs implement algorithms. [P3]	Students must communicate solution ideas within groups and as part of class discussion, as well as in Reflection Questions.
5.3.1 Evaluate a program for correctness. [P4]	Students test and debug their own code and evaluate others' in the Reflection Questions.
5.3.2 Develop a correct program. [P2]	Programmed solutions to challenges must work.
5.3.3 Collaborate to solve a problem using programming. [P6]	Students develop solutions in teams.
5.4.1 Employ appropriate mathematical and logical concepts in programming. [P1]	Relationships such as "distance per wheel rotation" are important to making solutions work.
7.4.1 Connect computing within economic, social, and cultural contexts. [P1]	Reflection Questions ask students to make evaluative recommendations based on the impacts of robotic solutions in context.