Introduction

The RoboMaster EP CORE Curriculum

The RoboMaster EP CORE robot is a physical computing solution offering the opportunity to combine computing, robotics, engineering, and artificial intelligence. The standards aligned curriculum has been mapped out to ensure students are introduced to concepts, investigate real world examples of application, and engage in a practical application of the skills learned.

RoboMaster EP CORE is a robust and engaging robot and lends itself to a project based learning pedagogy, allowing students the opportunity to engage in real world examples through investigation with authentic and engaging learning experiences. The lesson sequence allows skills to be developed including opportunities for further enhancements through personalized learning aimed at differentiating content for all student's strengths and needs.

The RoboMaster EP CORE

With an integrated powerful CPU, the Intelligent Controller can simultaneously support functions such as low-latency high-definition image transmission, AI computing, and programming development. It also coordinates transmission seamlessly to execute command signals.

The RoboMaster has been constructed to maximize opportunities for students with Mecanum wheels, each with 12 rollers allowing omnidirectional movement and precision control. In addition, there is front axes suspension allowing for added versatility during operation. A gimbal with yaw and pitch rotation offers an expansive field of view for the FPV (First Person View) camera. It is complimented with six intelligent armour panels which cover the robot integrating sensors into the program design.

Lesson structure and management

Within each lesson, students are engaged with a precise sequence of activities to allow students to progress with their computing skills, knowledge and understanding.

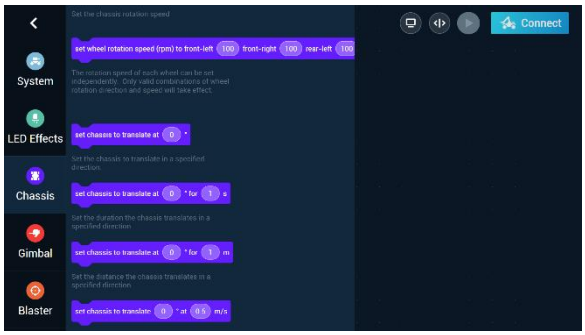
The lessons start with a warmup to engage students in the learning intended within the lesson. This is delivered by the teacher with active input from students. The learn section contains the theory element to develop the knowledge required to follow through to the application of the skills learned within the block-based programming environment. The theory is delivered by the teacher with opportunities for discussion and student input. The practical application is student driven with the teacher facilitating, where required, to aid understanding and development. The final element is to wrap up the lesson with activities to reflect and assess students' progress.

Each lesson is designed to take 60 minutes with the opportunity to expand with free coding experience at the end of a lesson to build upon the lesson skills, knowledge and understanding. The utilization of open-ended challenge cards ensures students requiring stretch activities can continue to develop past the lesson content. The utilization of student code support handouts ensures block code is available to students where required.

Before the class starts, it is recommended that an open space is acquired to ensure groups have enough space for the practical application of the lesson. Group numbers can be defined to meet classroom requirements with

two or four being a suitable number to maximize practical application stages. Teachers will ensure that iPad's or PC's are prepared for a group of two or four. The RoboMaster App on the PC can be used to give teacher guidance to support difficulties students could encounter.

The RoboMaster platform is intuitive and engaging allowing ease of connection to the RoboMaster EP CORE and block-based programming language for all activities.



The programming blocks are organized in easy to identify sections with the ability to switch to the textual programming language Python and observe how the program can be written instead of utilizing programming blocks.

Each lesson pack contains:

- Detailed lesson plan
- Student facing slides
- Student code support handout
- Challenge cards for stretching students further
- Activity resources where required

Students will be given the opportunity to utilize flowcharts as part of program planning. They will be encouraged throughout lessons to create and complete effective testing tables to allow debugging and improvements to be made. Integrating peer feedback and collaboration within groups will allow students to build on their own skills, knowledge and understanding through peer support.

The main aim of the RoboMaster EP CORE curriculum is to harness a student's passion for computing, robotics, engineering, and artificial intelligence and apply it to a series of thought provoking, authentic activities to develop active learners.

Grade 6 Syllabus

The focus of grade 6 will be to develop understanding of the hardware of the robot and be able to start to create simple programs to control the hardware on the robot, like LED lights and movement.

Lesson	Learning Objectives
1 Robot Basics	To identify the 3 laws of robotics To identify the main components of the robot To create a program to move the robot in a cross-shaped track on the ground
2 Sequence /Debugging	To identify the shapes to build a basic flowchart To create a flowchart to demonstrate a sequence within a program To create a program to turn all the gimbal lights on counter clockwise
3 Repetition	To create a flowchart to demonstrate repetition To describe what is meant by a nested loop To create a program to allow the robot's LED lights to blink
4 Selection	To create a flowchart to demonstrate selection To describe how selection is used within a program To create a program to realize the random shifting between two lighting effects
5 Computational thinking	To define the computational thinking concepts To discuss the importance of commenting in code To demonstrate decomposition and commenting with a given program
6 Variables and data types	To identify the difference between a controller and microcontroller To describe what a variable is and how it stores different data types To create a program utilizing a variable
7 Smart technology	To identify devices that harness smart technology To discuss how sensors are used within smart technology To create a program utilizing a sensor
8 The role of the actuator	To identify the role of the actuator To discuss the hardware within the actuator of the robot To create a program to check a single motor and a program to check all motors
9 Compound conditionals	To describe what a compound conditional is To create a flowchart to demonstrate a compound conditional To create a program utilizing a compound conditional

CSTA Standards Aligned

Standards Code	Lesson 1	Lesson 2	Lesson 3	Lesson 4	Lesson 5	Lesson 6	Lesson 7	Lesson 8	Lesson 9
2-CS-01	Y	Y	Y	Y	Y	Y	Y	Y	Y
2-CS-02	Y	Y	Y	Y	Y	Y	Y	Y	Y
2-CS-03	Y	Y	Y	Y	Y	Y	Y	Y	Y
2-NI-04									
2-NI-05									
2-NI-06									
2-DA-07									
2-DA-08							Y		
2-DA-09							Y		
2-AP-10		Y	Y	Y					Y
2-AP-11						Y			
2-AP-12			Y						Y
2-AP-13	Y	Y	Y	Y	Y	Y	Y	Y	Y
2-AP-14									
2-AP-15									
2-AP-16	Y	Y	Y	Y	Y	Y	Y	Y	Y
2-AP-17	Y	Y	Y	Y	Y	Y	Y	Y	Y
2-AP-18									
2-AP-19					Y				
2-IC-20									
2-IC-21									
2-IC-22									
2-IC-23									

NGSS Standards Aligned

Engineering design

MS-ETS1-1
Define the criteria and constraints of a design problem with sufficient precision to ensure a successful solution, taking into account relevant scientific principles and potential impacts on people and the natural environment that may limit possible solutions.

MS-ETS1-2
Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.

MS-ETS1-3
Analyze data from tests to determine similarities and differences among several design solutions to identify the best characteristics of each that can be combined into a new solution to better meet the criteria for success.

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.

Grade 7 Syllabus

The focus of grade 7 will be to further develop their understanding of the hardware of the RoboMaster EP CORE. Students will be able to create more complex programs integrating the use of the RoboMaster hardware including: sensors and Gimbal.

Lesson	Learning Objectives
1 Gimbal	To define what a Gimbal is used for To identify the Gimbal on the RoboMaster EP Core To create a program to move in a V-shape and a circle
2 Sensors	To define what a sensor is and how they are used To identify the sensors built into the RoboMaster EP CORE To create a program to output angle measurement
3 Passwords	To define the purpose of a function in a program To discuss the physical and digital security measures available To create a program utilizing functions
4 Artificial Intelligence basics	To define what artificial intelligence is To identify the difference between a robot and artificial intelligence To create a program that reacts to the number of claps heard
5 Artificial intelligence application	To list a range of applications for artificial intelligence To discuss the advantages and disadvantages of artificial intelligence To create a program that reacts to a person or gesture
6 Image recognition	To define how an image is recognised To discuss how vision markers are used on RoboMaster EP CORE To create a program using vision markers to control movement
7 Control system basics	To identify the different stages of the input-process-output process To define what a control system is To create a program to use vision markers for keeping a distance optimization
8 Control system Open and closed loops	To define the difference between an open and closed loop control system To discuss the advantages of automatic tracking using vision markers To create a program that will follow vision markers
9 Skill Application	To define the roles associated with pair programming To discuss the advantages and disadvantages of pair programming To create a program demonstrating pair programming

CSTA Standards Aligned

Standards Code	Lesson 1	Lesson 2	Lesson 3	Lesson 4	Lesson 5	Lesson 6	Lesson 7	Lesson 8	Lesson 9
2-CS-01	Y	Y	Y	Y	Y	Y	Y	Y	Y
2-CS-02	Y	Y	Y	Y	Y	Y	Y	Y	Y
2-CS-03	Y	Y	Y	Y	Y	Y	Y	Y	Y
2-NI-04									
2-NI-05			Y						
2-NI-06									
2-DA-07									
2-DA-08		Y							
2-DA-09		Y							
2-AP-10	Y	Y	Y	Y	Y	Y	Y	Y	Y
2-AP-11	Y	Y	Y				Y	Y	Y
2-AP-12						Y			
2-AP-13	Y	Y	Y	Y	Y	Y	Y	Y	Y
2-AP-14			Y	Y	Y				
2-AP-15									
2-AP-16	Y	Y	Y	Y	Y	Y	Y	Y	Y
2-AP-17	Y	Y	Y	Y	Y	Y	Y	Y	Y
2-AP-18									
2-AP-19									
2-IC-20									
2-IC-21				Y	Y				
2-IC-22									Y
2-IC-23			Y						

NGSS Standards Aligned

Engineering design

MS-ETS1-1
Define the criteria and constraints of a design problem with sufficient precision to ensure a successful solution, taking into account relevant scientific principles and potential impacts on people and the natural environment that may limit possible solutions.

MS-ETS1-2
Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.

MS-ETS1-3
Analyze data from tests to determine similarities and differences among several design solutions to identify the best characteristics of each that can be combined into a new solution to better meet the criteria for success.

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.

CSTA Standards Key

Standards Code	Level 2 (Ages 11-14) By the end of Grade 8, students will be able to…
2-CS-01	Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices
2-CS-02	Design projects that combine hardware and software components to collect and exchange data.
2-CS-03	Systematically identify and fix problems with computing devices and their components.
2-NI-04	Model the role of protocols in transmitting data across networks and the Internet.
2-NI-05	Explain how physical and digital security measures protect electronic information.
2-NI-06	Apply multiple methods of encryption to model the secure transmission of information.
2-DA-07	Represent data using multiple encoding schemes
2-DA-08	Collect data using computational tools and transform the data to make it more useful and reliable.
2-DA-09	Refine computational models based on the data they have generated.
2-AP-10	Use flowcharts and/or pseudocode to address complex problems as algorithms.
2-AP-11	Create clearly named variables that represent different data types and perform operations on their values.
2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals
2-AP-13	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
2-AP-14	Create procedures with parameters to organize code and make it easier to reuse.
2-AP-15	Seek and incorporate feedback from team members and users to refine a solution that meets user needs.
2-AP-16	Incorporate existing code, media, and libraries into original programs, and give attribution.
2-AP-17	Systematically test and refine programs using a range of test cases.
2-AP-18	Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.
2-AP-19	Document programs in order to make them easier to follow, test, and debug.
2-IC-20	Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options
2-IC-21	Discuss issues of bias and accessibility in the design of existing technologies.
2-IC-22	Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.
2-IC-23	Describe tradeoffs between allowing information to be public and keeping information private and secure.